Effect of Open Student Models on Self-assessment, Problem Selection and Learning

A thesis submitted in partial fulfilment of the requirements for the Degree

of Master of Science in Computer Science

in the University of Canterbury

by Mohammad Ahmadi

Prof. Antonija Mitrovic (Supervisor),

Dr. Moffat Mathews (Associate Supervisor)

University of Canterbury

2017

Table of Contents

List of Figures

List of Tables

Acknowledgments

First allow me to thank my supervisor Professor Antonija Mitrovic, who supported me throughout the research period. She was there whenever I did not understand, and guided me whenever I was lost. She has always been so enthusiastic and patient to me and I really appreciated that. Completing this research and writing up the thesis has not been an easy thing for me. Without her efforts I could never reach this point. Special thanks for letting me to start my degree in New Zealand which is the start of a new chapter of my life.

I also want to thank my co-supervisor Dr. Moffat Mathews, special thanks for his guidance and support throughout this research.

Thanks to ICTG fellows with their strong support, especially Jay Holland. I must thank Enos Chen and Geela Venise Fabic for helping me in conducting the classroom study and also thanks to all previous contributors to EER-Tutor and ICTG members. I am very proud to be one of them.

Finally, I would like to thank my sister who helped me and supported me to study overseas. Without her financial support I could not finish this degree.

Abstract

Student modeler, as the central component of Intelligent Tutoring Systems (ITSs), has been formed to assist in systems' decision making for students' learning. The ITS can adapt its pedagogical actions to provide personalized learning feedback by analyzing students' knowledge represented in the student modeler. It is well-known that viewing individual Open Student Models (OSMs) can help students to reflect on their own learning progress and enhance their meta-cognitive skills, such as self-assessment and problem selection. It is also shown that better meta-cognitive skills lead to better learning outcomes. By knowing their strengths and weaknesses in the corresponding domain through inspecting the OSM, students can develop a more effective and efficient way of learning.

On one hand, the OSM can provide detailed information about the student state of knowledge. On the other hand, it is important for any instructional method to effectively and efficiently utilize the limited human working memory which directly impacts the design of OSM. It is shown that the performance of instructional systems can drop because of the under load or overload of the learner's cognitive capacity. Our aim here is to study the effect of the type and amount of information presented to student in their OSM on their learning outcome, self-assessment skills and their problem selection skills as well as the motivation to utilize these meta-cognitive skills.

We picked a problem-solving environment called EER-Tutor, which is a web-enhanced ITS that supports university students in learning conceptual database modelling using the Enhanced Entity-Relationship model (EER), as our test bed for our study. We designed a new strategy for presenting information about the student's progress via OSM and problem selection page, and we evaluated the impact of this new presentation on student's learning and meta-cognitive skills by running a classroom experiment.

1 Introduction

Acquiring new knowledge and skills, known as learning, is an important part of everyone's life. Growing technology has resulted in a rapid increase in the development of educational tools and methods which help in enhancing learning. In the last three decades, various types of tutoring systems have been introduced as tools to help students in their studies.

It is well understood that because students have different background knowledge, experiences, personal abilities and motivations, the same learning environment and support does not result in the same learning effectiveness (Holt, Dubs, Jones, & Greer, 1994). Therefore, many research projects have been conducted in the area of individualization of learning process.

1.1 Artificial Intelligence in Instructional Systems

One-to-one tutoring is shown to be more effective than conventional learning in classrooms (Bloom, 1984). The average student in a one-to-one tutoring session is shown to perform about two standard deviations above the mean of a conventional class (Bloom, 1984). But human one-to-one tutoring is not practical or affordable in most cases, which motivated research in the area of computer-based tutoring that can mimic a human tutor by adjusting their feedback based on the learner's needs.

Researchers used Artificial Intelligence (AI) to develop computer-based instructional systems that can help with bridging the gap between conventional classrooms and one-to-one tutoring. With the help of AI, these instructional systems can provide individualized learner support which is a computationally complex task.

Computer-Based Tutoring (CBT) and Computer-Aided Instruction (CAI) are the very first types of Computer-Based tutoring systems developed in the 1950s where the education content of the tutoring system was statically defined. The 1960s generation of CAI systems, known as branching programs, only compared student's answers to predefined solutions but could not provide adaptive learning support to match the learner's individual needs.

The next generation of computer-based tutoring systems, also known as Intelligent Tutoring Systems (ITS), first appeared in the 1970s. ITSs are intelligent so that they can provide individualized learning and customized feedback to learners. These systems attempt to reproduce and stimulate different aspects of an expert human tutor. The development of such ITSs is an interdisciplinary field of research, which includes artificial intelligence, computer science, education, and psychology.

1.2 Research Overview

Student modeler is the fundamental component of ITS which provides information to the pedagogical component to produce personalized actions for each learner. The student modeler forms the student's state of knowledge in the instructional domain which is the basis of the adaptive learning support provided by ITS.

Traditionally, the student model was not accessible by the student and it was only used by the system to generate personalized feedback and instruction for the learner. Sharing this student model with the student, also known as Open Student Model (OSM), was an innovative idea which was explored in recent years by researchers (Kay, 2008). OSMs are shown to have positive impact on student learning outcome (Bull, 2004; Bull & Nghiem, 2002; Mitrovic & Martin, 2002).

It is also shown that OSM can have positive impact on helping student to learn how to learn and improving their meta-cognitive skills such as self-assessment and problem selection (Mitrovic & Martin, 2007). When OSM presents the weakness and strength of the student in domain knowledge, the student can use this information to reflect on their learning process and adjust their learning strategies. The goal of ITS is not only to support personalized learning in a knowledge domain but also to teach student how to learn effectively and efficiently.

In some OSM, the learner does not receive any support from the ITS in regard to how to select problems in a problem-solving environment. In other words, this type of OSM only provides information about the state of knowledge (correct and incorrect knowledge gained by student) and the learner has the full control over their learning strategies and process. Thus, it is very important for the design of such OSM to include aspects which helps student to develop meta-cognitive skills such as self-assessment and problem selection skills (Mitrovic & Martin, 2007).

The OSM can provide detailed information about the student state of knowledge but the Cognitive Load Theory (CLT) states that humans have a limited working memory (Miller, 1956) but comparatively unlimited long-term memory and thus it is important for any instructional method to effectively and efficiently utilize these resources while designing the OSM. The working memory has the limited capacity of holding seven plus or minus two chunk of information (Miller, 1956). It is shown that the performance can drop because of the under load or overload of learner's cognitive capacity (Young & Stanton, 2002).

Our aim is to study the effect of the type and amount of information presented to student in their OSMs on their learning outcome, self-assessment skills and their problem selection skills. In our research, we evaluated this effect by conducting a classroom experiment.

We selected a problem-solving environment called EER-Tutor, which is a web-enhanced ITS that supports university students in learning conceptual database modelling for Enhanced Entity-Relationship (EER) that is developed by the Intelligent Computing Tutoring Group (ICTG) at the University of Canterbury (Suraweera & Mitrovic, 2004), as our test bed for our study. We modified the existing EER-Tutor OSM to have two types of views with different number of concepts about the knowledge domain. We hypothesized that having access to an OSM with more detailed concepts will help students understand the knowledge domain better and also gain better self-assessment skills which can lead to better learning outcome.

We also changed the problem selection page of the EER-Tutor so that it presents problems categorized based on their complexity instead of the original flat organization. In our new problem selection page, the progress of student on each level of problem complexity is also displayed. We hypothesized these changes will help student to make more informed decisions toward selecting the next problem which can eventually lead to more efficient learning and better learning outcome.

The evaluation part of this research develops measures to quantify and qualify our hypothesis. The results will guide us to future steps of design and development in this area.

1.3 Thesis Structure

This thesis consists of six chapters. Chapter 2 presents background information and literature review relevant to this research project. This chapter consists of seven sections: presenting an overview of ITSs and its components, approaches to student modeling and Open Student Model (OSMs) and OSM's different visualizations. The following section provides details of EER-Tutor and its student model and OSMs. The next section describes Cognitive Load Theory (CLT) and instructional efficiency (E). The last section is focused on problem selection and theory of mastery.

In Chapter 3, we discus some related work to this thesis and our motivation of doing this research and then we present our objectives and research hypotheses.

Chapter 4 focuses on the design and implementation. In this chapter, we explain our strategy for presenting information about the student progress in two different OSM, Skill meter and Concept list, in EER-Tutor and what was added to these two OSMs is described in great detail. We also explain the changes made to the problem selection page and the problem categorization into five different levels based on the complexity of the concepts they cover.

Chapter 5 presents the evaluation study, including how the evaluation has been conducted, what data have been collected and results. Some suggestions are provided for the future enhancements and some ideas for future directions of this research is presented.

Chapter 6 summarizes the overall work of this research. A conclusion is drawn based on the findings from evaluation.

2 Literature Review

The objective of this chapter is to lay out the background of our work by presenting some related works and findings. We will first describe the architecture of Intelligent Tutoring Systems (ITS) and then provide details about their components. Different student modelling approaches are then introduced. Open student models (OSM) and their benefits and different representations are explained next. The details on EER-Tutor, which is the test bed for this research project, are provided next. The student model part of EER-Tutor is highlighted since it is where we made modifications. In the last section, we discuss the Cognitive Load Theory (CLT) and its measures to evaluate instruction. We also elaborate on self-assessment and problem selection skills as metacognitive skills enhanced by OSM.

2.1 Intelligent Tutoring Systems

Learning is the process of acquiring knowledge in a subject. The need of each individualized learner in this process greatly depends on their background knowledge in that subject. The one-to-one human tutoring which can adapt dynamically to this individual need is more effective than traditional classroom but the first generation of computer assisted tutoring systems did not support individualized learning. Therefore, the goal of the next generation of computer-based tutoring systems, also known as ITS, was to simulate different aspects of an expert human tutor.

The goal of ITSs is not only to provide individualized tailored feedback and help in learning a particular domain, but also to teach students how to efficiently gain that knowledge. In other words, ITS also aim to help students develop meta-cognitive skills. Meta-cognition generally includes any activity that involves inspecting and adjusting learning approaches and habits. The following are listed by Self (1995) as meta-cognitive activities: explaining something to self or others, self-assessment, planning appropriate learning strategies, monitoring, allocating resource, recognizing problems. Research in several disciplines has shown that meta-cognitive skills can be taught (Bielaczyc, Pirolli, & Brown, 1995).

Meta-cognitive skills such as self-assessment are critical for efficient learning. Better self-assessment helps student to know where their weaknesses are, so they can focus on improving them. That also helps them to decide if a question is too hard and they should abandon it or continue working on it. Thus it is important for any ITS to support the acquisition of meta-cognitive skills, especially self-assessment skills. Swanson (1990) showed that better learning and improved problem solving result from better meta-cognitive skills.

The development of ITSs is an interdisciplinary field of research composed of artificial intelligence, computer science, human-computer interaction, education, statistics, and cognitive psychology. Since the process of learning happens in human mind, there has been a significant amount of research focusing on cognitive psychology to analyze human skill acquisition. Cognitive architecture, which is known as the fundamental of cognitive psychology, analyses how human mind works. The goal of ITS is to benefit from this analysis toward providing individualized instructions to students on how to perform a task. ITSs have been developed for a wide range of educational domain such as algebra, geometry, physics, computer science and programming languages (Anderson & Lebiere, 1998).

2.2 Components of ITSs

Different ITSs have used different approaches in their development, but they all normally contain these four fundamental components: Pedagogical Module, Domain Model, Student Model and Interface (Freedman, Ali, & McRoy, 2000; Polson & Richardson, 1988).

The **Pedagogical Module** includes an instructional technique that controls the tutoring process, for example the level and type of the support. The domain module and student model provide data to the pedagogical module as inputs. The aim of the pedagogical module is to individualize support based on the student’s background knowledge and generate appropriate feedback when mismatches between students’ knowledge and the domain expert’s knowledge are found. Then the pedagogical actions can be forwarded to the user interface.

The **Domain Model** includes the knowledge the tutor is teaching. Domain or expert module has to contain accurate and detailed knowledge which is accessible to other components to use and is generated by experts who have strong experiences in a particular domain (Polson & Richardson, 1988).

The **Interface** includes methods of interfacing between the learner and ITS. Interface design requires thinking of how the material should be presented to the learner in the most effective way.

The **Student Model** records individualized information about student’s knowledge such as their understanding and misconceptions by analyzing the students’ interaction with the system. Student model is where the system keeps track of student's learning gain and knowledge in each concept. Pedagogical feedback is also derived based on the information stored in student models.

Since the focus of this research is on different aspect and views of student modelling and their impact on learning, the following sections will describe different student models and their characteristics as well as their advantages.

2.3 Student Modelling

Student modeler is a fundamental part of an adaptive and robust tutoring system, since individual pedagogical actions for each learner are decided based on the student modeler. ITS collects data about all students’ activities in the learning environment, such as students’ performance on problem solving and their interactions with the system. The student modeler then analyzes these data to model each student’s state of knowledge in the instructional domain. The individualized pedagogical feedback is generated based on the student’s state of knowledge. Thus the main challenge in successfully modelling a student is to construct a complete and accurate model of student's knowledge (Self, 1990). A complete model should not only include what a student knows correctly but also what a student knows incorrectly. Building such a model can be a very difficult task since the range of misconception is unlimitedly wide.

Student model usually maintains the representation of a student performance and his/her knowledge state in the domain by compiling the relevant information acquired by system about the student. Many approaches are suggested in literature for student modelling for example *Overlay models*, *Perturbation models*, models based on *Machine Learning*, *Model Tracing* and more recently, *Constraint-based models* (Holt et al., 1994; Mitrovic, Koedinger, & Martin, 2003; Mitrovic, Martin, & Mayo, 2002; Ohlsson, 1994).

Overlay models rely on the fact that the student knows a subset of all the concepts that exist in the domain model and thus the goal of ITS is to help student to learn the missing domain knowledge in order to master that domain (Carr & Goldstein, 1977).

Perturbation models assume that the student may have misunderstanding of concepts, also known as buggy knowledge, which also should be reflected in the student model (Holt et al., 1994). Thus the domain model of a perturbation model usually includes bug libraries which are libraries of mistakes students normally make. The formations of these libraries are normally an extensive task and performed by studying student's behavior. But these libraries are not transferrable between different groups of students since they may not cover all mistakes made by another group (Payne & Squibb, 1990).

Machine learning models uses machine learning algorithms such as ID3 and PRISM to actively generate the student model online by searching the set of possible models (Gilmore & Self, 1988).

The most important challenge of student modelling is how to form and maintain a realistic model for each student. This can be an intractable task and since this search space is huge, the task of modelling a student can be very complicated.

2.3.1 Model Tracing and Knowledge Tracing

The process of acquiring a new skill is considered as going through three stages: cognitive stage, associative stage and autonomous stage (Fitts, 1964). In the cognitive stage, knowledge is declarative and need interpretation thus it is slow and buggy (Anderson, 1982). But knowledge in the autonomous stage is procedural knowledge which is compiled, fast and error-free. The stage in between is associative stage when the knowledge is partially declarative and partially procedural/compiled (Anderson, 1982).

Procedural knowledge is a set of procedures that are also called production rules. Human actions are a sequence of smaller sequential and coordinated actions which are aimed for an actor goal and are adapted to the situation they occur. This is the basis of production rules and can be summarized as:

Rule: Goal, Situation → Action.

This means the human brain constantly looks for the rule that is relevant to the current goal and situation to decide which rule to execute. In other word, the human brain matches the situation and goal of all rules against the current situation and goal and executes the matching rule. If there is more than one matching rule, the declarative memory decides the order of execution.

Cognitive architecture claims that human brain performs cognitive tasks by computing. The ACT-R theory (Anderson, 1993) of cognition, based on the cognitive architecture, claims that cognitive skills are acquired from learning and following (computing) a specific set of production rules. Cognitive models teach production rules by reacting to each step that students take to solve a problem. Thus a cognitive model can be modeled as a system of if-then production rules which are the basis of correct steps and incorrect steps. This cognitive model is the basis for two student modeling techniques: model tracing (short-term modelling) and knowledge tracing (long-term modelling).

Model tracing monitors student's progress through a problem solution. In model tracing, the student’s steps are compared against a set of rules form the model (Anderson, Corbett, Koedinger, & Pelletier, 1995). This model includes correct and incorrect rules (buggy rules). A mistake is detected if a student's step matches a buggy rule or does not match any rule. Knowledge tracing monitors students' learning from problem to problem (Corbett & Anderson, 1992). Students' strengths and weaknesses relative to the production rules are tracked by this model.

2.3.2 Constraint-Based Modelling

Self (1990) suggested that it is not necessary to model everything you can about the student, instead the student model should be built so that it can support the pedagogical actions that ITS wishes to take. In other words, a student model does not have to be complete and accurate to be useful. The fact is even human teacher only know very loose models of their students but yet they are highly effective in teaching (Ohlsson, 1994). This observation is the basis of Constrain Based Modelling (CBM) approach which offers reduces complexity of student modelling.

CBM is based on Ohlsson's theory of learning from performance errors (Ohlsson, 1996) and it introduces a way of overcoming the intractable nature of student modelling. In CBM, the domain knowledge consists of a set of explicit constraints on correct solutions which also implicitly represent the incorrect solutions (Mitrovic & Ohlsson, 1999). Thus these constraints divide all solutions into two groups: correct and incorrect. The fact that CBM does not require extensive studies to form bug libraries makes it a lot less complex comparing to cognitive tutors (Ohlsson, 1994).

A constraint is an ordered pair $<C_r, C_s>$, where C_r is the relevance condition determining a set of problem state in which the constraint is relevant, and C_s identifies the state in which the constraint is satisfied. If a constraint is relevant in some state, then it must also be satisfied in order for the solution to be correct, otherwise the solution contains an error. Thus, the semantics of a constraint are: if the C_r condition is true, then the C_s must also be true, otherwise something has gone wrong.

A simple example in daily life would be: If you drive a car on the road, you should drive on the right side of the road. In this example, the relevant condition is met when the person is driving a car on the road, the correct state requires the satisfaction condition to be met too, which means the car should be on the right hand side of the road.

Constraints can be divided to two types: constraints representing syntactic properties of knowledge domain which only relate to student's solution and constraints that represent semantic properties of the domain which refer to relation between the student's solution and the ideal solution (Mitrovic, Mayo, Suraweera, & Martin, 2001).

In CBM, the evaluation of student solution is performed by comparing them against the constraint set. Starting with the relevance condition, if that matches with the problem state then the satisfaction condition is compared against the problem state. If the satisfaction pattern matches the state, the constraint is satisfied, otherwise it is violated. All satisfied and violated constraints are stored in short-term student model. All constraints used by student and their history are stored in long-term student model.

As discussed in (Mitrovic et al., 2003), MT and CBM are two major modelling techniques using cognitive tutoring and have their own strengths and weaknesses. MT is better suited for well-defined problem solving domains where comprehensive feedback is desirable. On the other hand, CBM is easier and faster to develop. MT focuses is on procedural knowledge but CBM focuses on declarative knowledge. MT tutors force student to follow a fix set of steps to solve a problem but CBM does not force any particular strategy to solve a problem.

SQL-Tutor (Mitrović, 1998; Mitrovic et al., 2001) for practicing Structured Query Language (SQL) in database management, and EER-Tutor (Suraweera & Mitrovic, 2004) for learning Enhance Entity-Relationship (EER) are designed based on CBM.

2.4 Open Student Model

Open Student Models (OSM) were proposed as a revolutionary idea in the area of personalized learning (Kay, 2008). In traditional personalized systems, student models were hidden from the students and only used by the system to make education process customized. Thus originally their purpose was to maintain a model to adapt tutoring systems to learning needs of individual students. On the other hand, it was argued by the pioneers of open student modelling that students can benefit from the ability to see and modify the state of their knowledge (Bull, 2004; Bull & Nghiem, 2002; Mitrovic & Martin, 2002).

Since the state knowledge in student model is accessible to students via OSM, they can review their understanding of each concept in the domain which in return increases their awareness of their knowledge level and encourages them to take charges in the learning processes (Bull, 2004; Bull & Nghiem, 2002). OSM provides student's progress over time and shows them how well and how much they have learned. Since they are aware of their strengths and weaknesses in the domain, they can optimize their own learning strategies. This helps the student to gradually develop meta-cognitive abilities (Mitrovic & Martin, 2007). Study in (Mitrovic & Martin, 2007) showed that OSM can significantly improve performance of novice students and self-assessment skills of advanced students.

OSM has been used in a variety of domain such as programming, mathematics, science, and second language. Opening various components of learning environment to the learner being modelled or sometimes also to other users and even the instructors has shown the following OSM benefits (Bull & Kay, 2010):

- Encouraging meta-cognitive activities such as self-reflection, planning and self-monitoring.
- Encouraging learner independence and taking control over their learning.
- Collaboration/competition among learners.
- Supporting interaction between learners, teachers.
- Supporting navigation between materials, problems and exercises.
- Providing formative assessment to students.

OSM is not simply showing the representation of student model from the system point of view to the student but instead it should be designed so that it can be understood by those who are still learning the subject. Thus it is important to design an effective interface for OSM (Bull & Kay, 2010).

2.4.1 Open Student Model Visualization

The OSM can use many different visualizations to indicate the knowledge level such as:

- **Arrows and targets**: Figure 1 shows an example of arrow and target. The numbers of arrows show the level of knowledge of a topic and the color intensity of the target shows the relevance of the topic to the current learning goal.
- **Smiling faces**: Figure 1 shows an example of smiling face where different smiling faces represent different levels of knowledge (mainly for younger learners).
- **Progress bar**: Figure 2 shows student's progress/misconception as bar for each concept in domain knowledge.
- **Color** is commonly used to indicate the knowledge level.
- **Text label and size** are also used to indicate the knowledge level.

Figure 1: Arrows and Targets in the left and smiling face on the right. Adapted from (Bull & Kay, 2010).

Skill meter: This model is one of the earliest and most common models because of their simplicity to understand and relative ease of implementation. The skill meter (or *Skillmeter*) uses progress bars to show the student's progress in concepts belonging to domain knowledge.

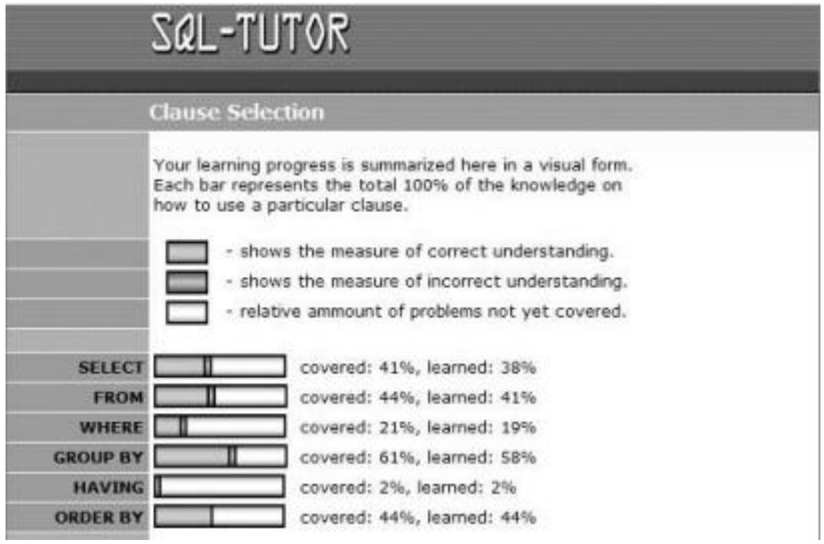


Figure 2: Skill meters in SQL-Tutor.

The skill meter normally includes information about the level of understanding, covered concepts and misconceptions. In some OSM the length of the meters implies the relevant size of the concept. Some OSMs show the probability of the user knowing the concept in text as well. Figure 2 and Figure 3 show some example of skill meters.

Concept list is an extension of skill meter. Comparing to skill meter the concept list has more concepts in knowledge domain. Figure 4 shows the Concept list in EER-Tutor.

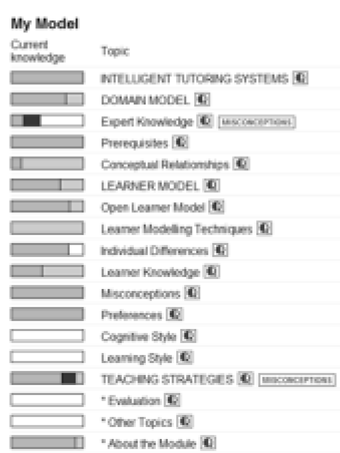


Figure 3: Skill meters in OLMlets. Adapted from (Bull & Kay, 2010).

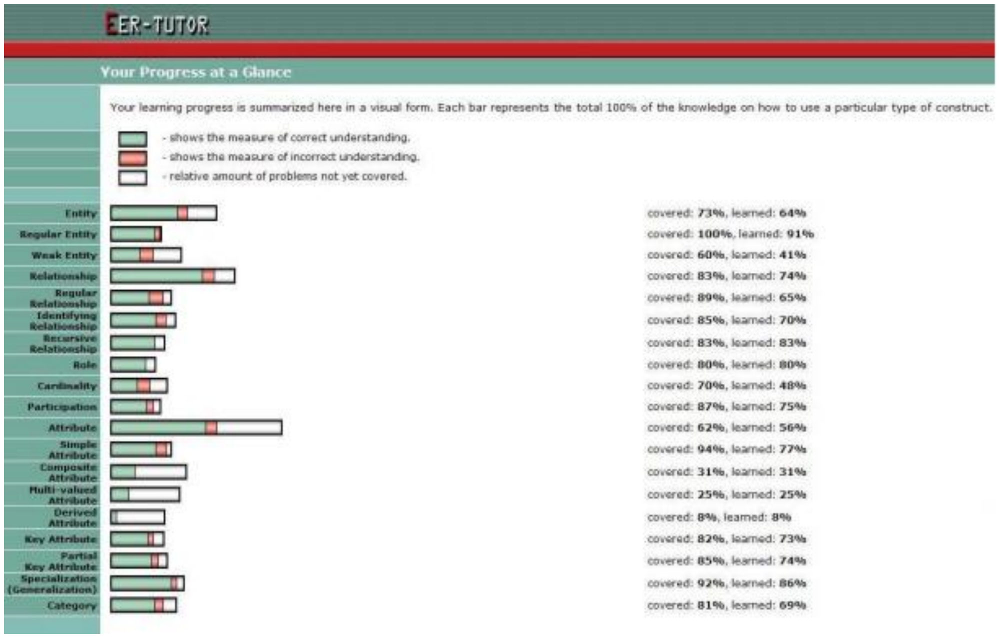


Figure 4: Concept list in EER-Tutor.

Concept maps is the most common structured OSM where the structure shows the domain reflecting either the learner's conceptual structure inferred by the system or constructed by the learner. Figure 5 and Figure 6 show two different concept maps.

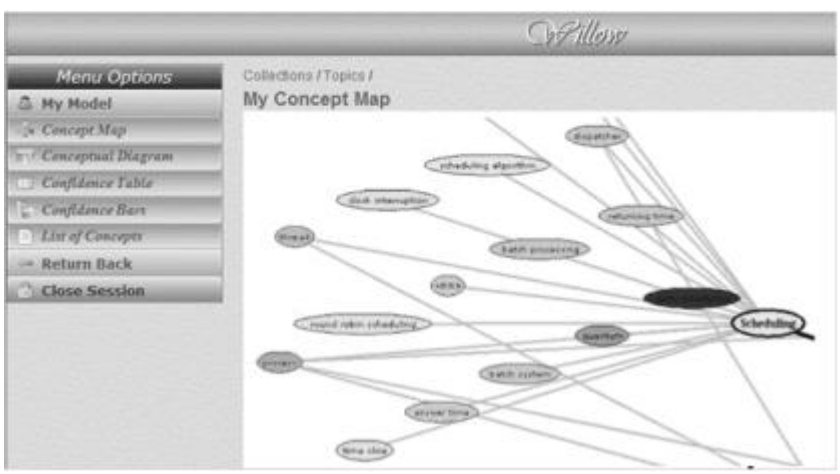


Figure 5: Concept map in Willow. Adapted from (Bull & Kay, 2010).

In Figure 6, one way arrows connecting the topics show their relationships. Since concept maps can carry more information about the knowledge domain in their structure, they can be harder to design and also harder to understand by the students but at the same time they can benefit from the advantage of encouraging students to learn the structure.

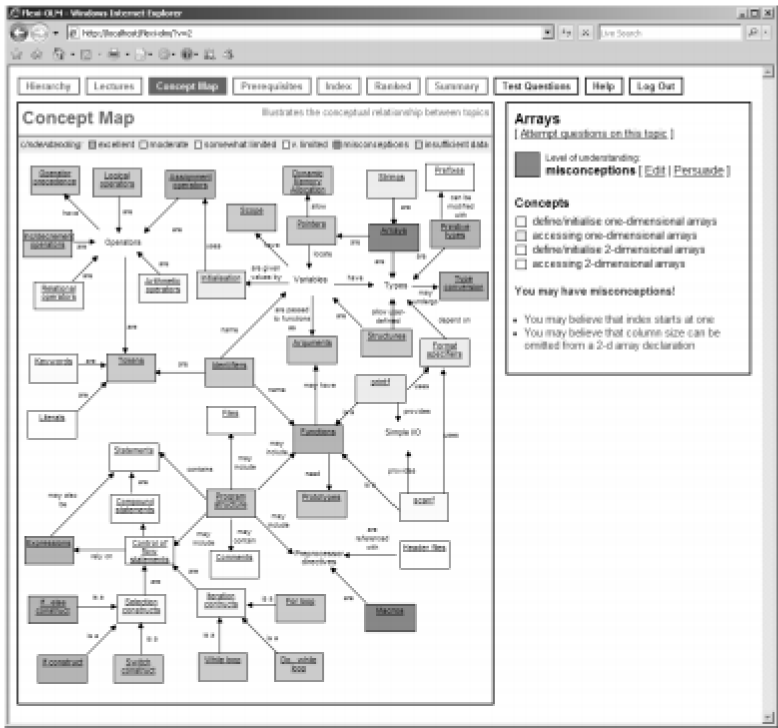


Figure 6: Concept map in Flexi-OLM. Adapted from (Bull & Kay, 2010).

Tree Structure: Some examples of other structured models are tree structure shown in Figure 7. Some tree structures allow the student to expand and contact nodes based on their needs.

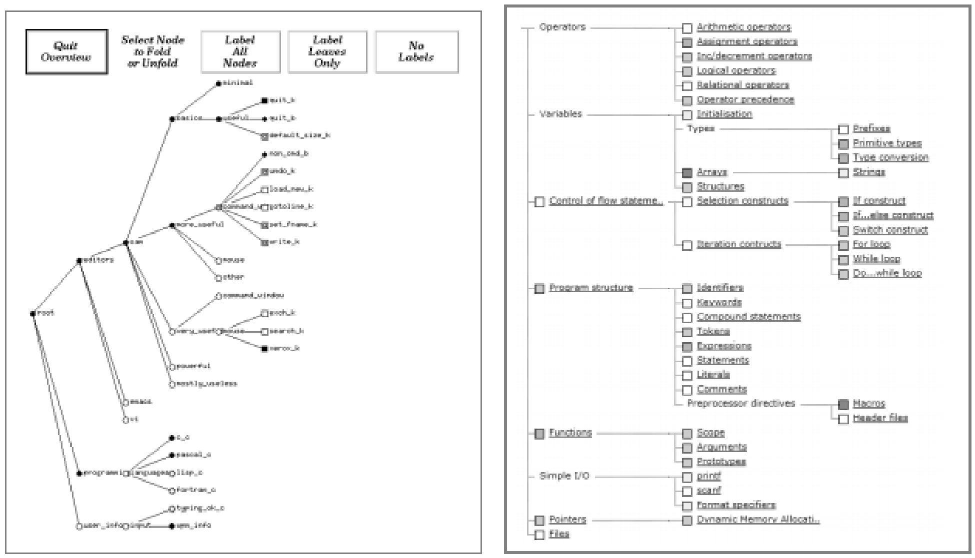


Figure 7: Tree structures in UM (left) and Flexi-OLM (right). Adapted from (Bull & Kay, 2010).

As shown in Figure 8, Concept Hierarchy is considered a tree based OSM view where concepts in domain form a tree structure. Thus in concept hierarchy the relationship between concepts are displayed using the tree. This information can help the student to understand the knowledge domain better.

The root of the tree includes the entire domain while sub-domains form the branches of the tree. The numbers in each rectangular represents the covered and learned percentage.

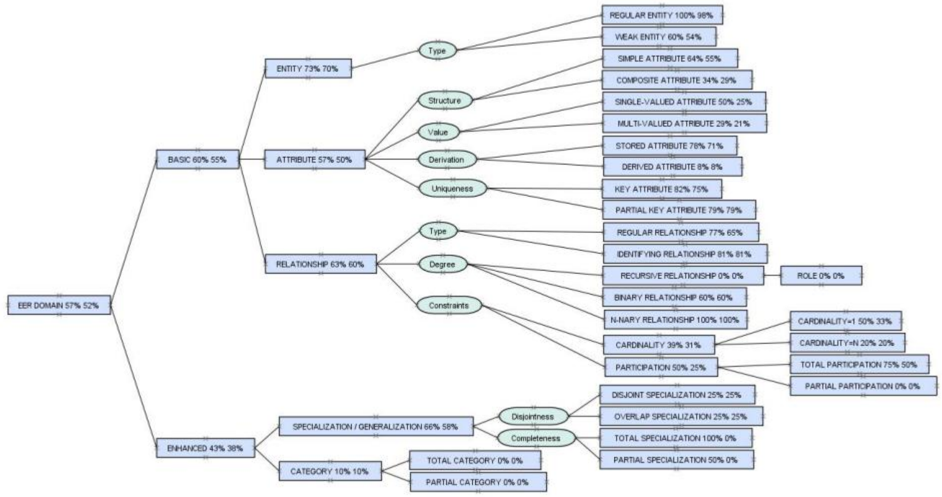


Figure 8: Concept Hierarchy in EER-Tutor. Adopted from (Mathews, Mitrovic, Lin, Holland, & Churcher, 2012).

Treemap: is a set of nested rectangles that form the visualization of the hierarchical data in a space-limited area. As shown in Figure 9, each rectangle in the treemap is equivalent to node in concept hierarchy. The size of each rectangle is proportional to the amount of material in the concept contained at that level of hierarchy. The color of each leaf represents the percentage of correct understanding.

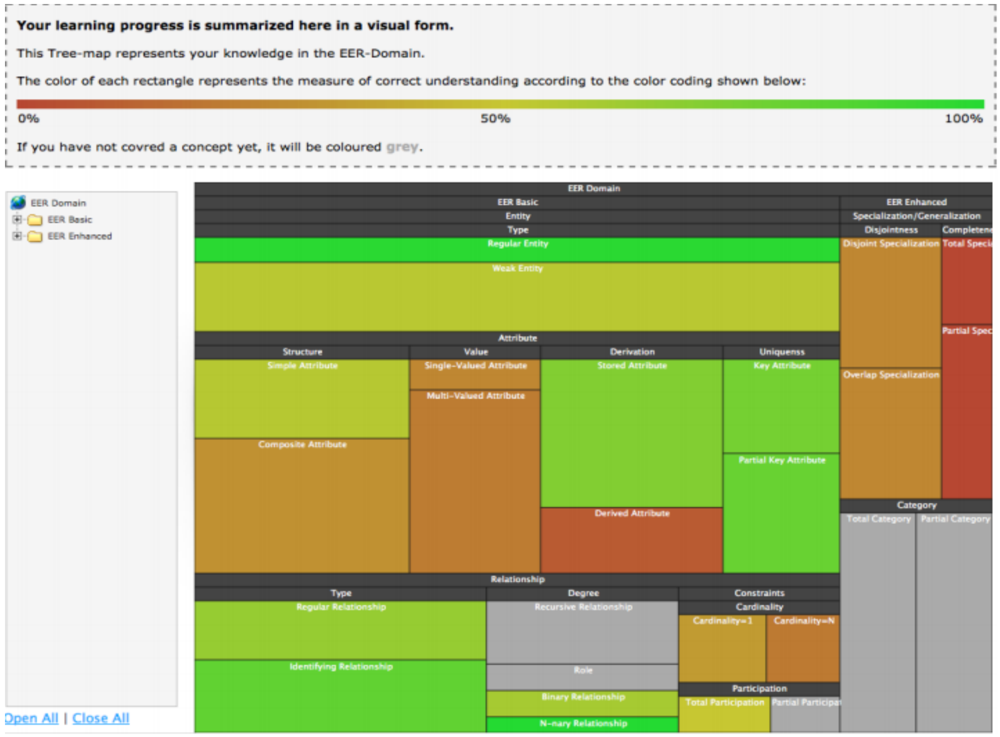


Figure 9: Treemap in EER-Tutor. Adopted from (Mathews et al., 2012).

Kiviat Chart: As shown in Figure 10, kiviat chart is a two dimensional chart where concepts form equal angle spokes starting from the same origin. The values displayed on each spoke are color coded to represent the amount of correct and incorrect understanding. This representation requires less effort from the students to understand only for a small set of concepts.

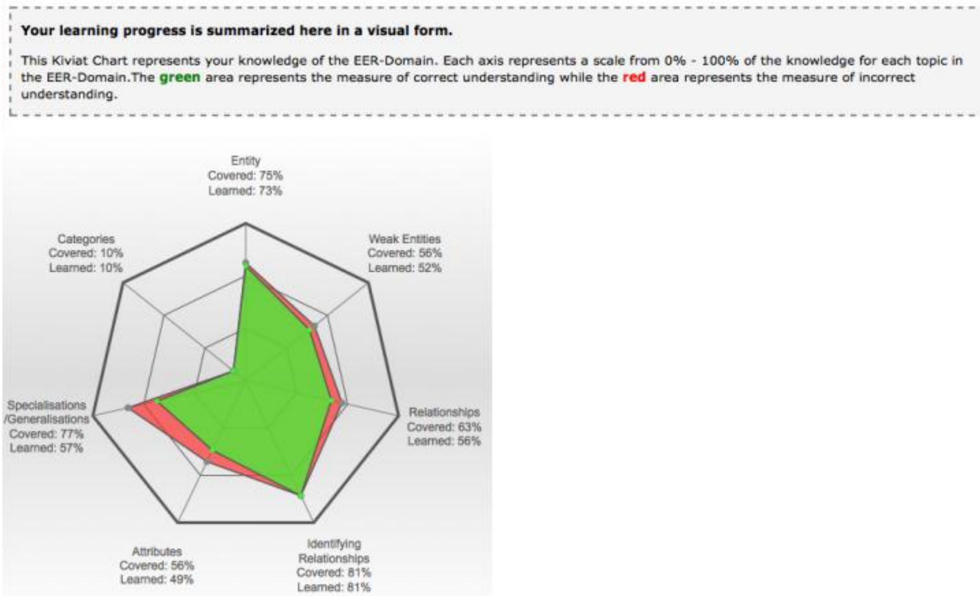


Figure 10: Kiviat Chart in EER-Tutor. Adapted from (Mathews et al., 2012).

Tag Cloud: As shown in Figure 11, tag cloud is formed by tags to describe all concepts in the domain. The size of the tag font shows the percentage of the concept covered. Color coding shows the measure of correct and incorrect understanding of the student in that concept.

Figure 11: Concept Tags in EER-Tutor. Adopted from (Mathews et al., 2012).

MasteryGrid: OSM can be extended to Open Social Student Model (OSSM), where learners have access to the model of the group. OSSM aims at improving the cognitive aspects of OSM by adding social aspect by letting students have access and explore each other models as well as cumulative model of the class. The idea of OSSM was initially suggested and studied by Bull (Bull & Kay, 2007; Bull, Mabbott, & Abu Issa, 2007). (Brusilovsky et al., 2015) investigate the added value of OSSM compared to OSM using MasteryGrid for Java programming in a single classroom and then for SQL programming in the larger scale.

Figure 12 shows the MasteryGrid interface. It shows the student knowledge organized by topic at the top row and allows the comparison of individual progress versus the progress of the class at the second and third row. Everything is colour coded to reflect the magnitude of the difference in progress. Student can select any topic to view detailed info on it, for each topic there are quizzes and examples.

MasteryGrid can also show anonymized rank list of individual student model like shown in Figure 13. These two studies showed that student motivation and student engagement were enhanced by OSSM compared to OSM and OSSM users worked more efficiently with the system. They also showed that learning gain of weaker students was significantly improved by OSSM.

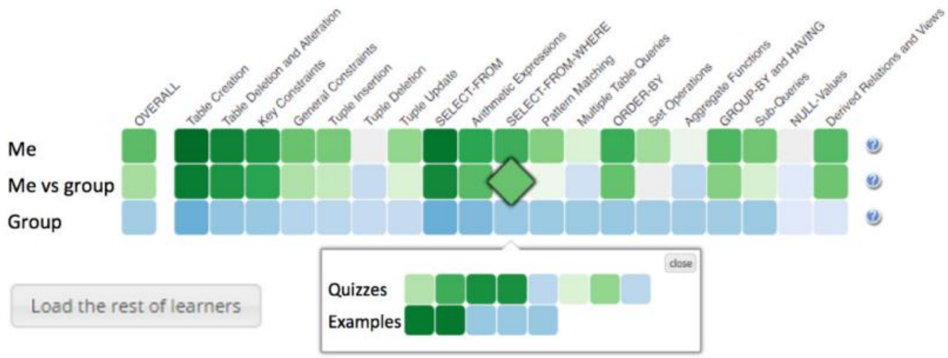


Figure 12: OSSM in MasteryGrid. Adapted from (Brusilovsky et al., 2015).

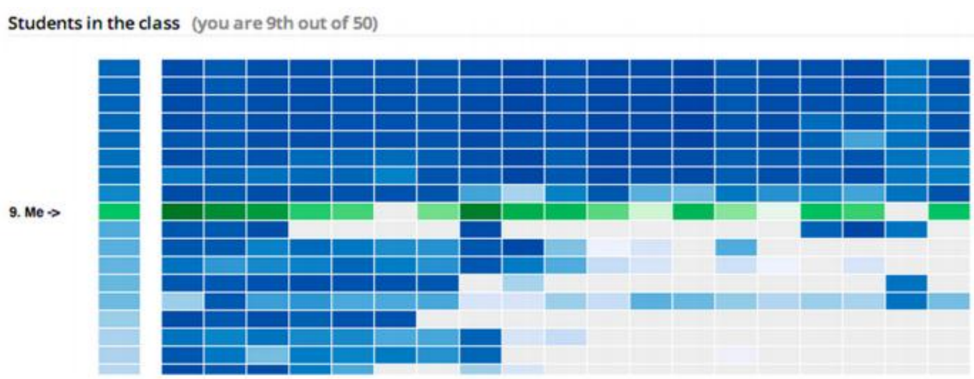


Figure 13: Anonymized ranked list of individual student models in MasteryGrid. Adapted from (Brusilovsky et al., 2015).

The student model can be opened not only to the user modelled but also to other users as well as instructors. Instructors can benefit from following the evolution of student's knowledge and so they can adapt their teaching to that. In some systems, the student can decide who can access their student model so they can allow their peers or even the system designer to see their model.

An Open Student Model focuses on helping students to find their strength and weaknesses themselves by inspecting the student model and then plan appropriate steps to overcome identified difficulties.

Thus enhancing meta-cognitive skills of student is very important in design of student models (Bull & Kay, 2008). Problem selection is one of the meta-cognitive skills that students need to acquire to manage their learning. OSM is shown to help students in making better choices about problems to solve and thus is proven to improve problem selection meta-cognitive skills (Kay, 2008).

2.4.2 Inspectable versus Editable Open Student Model

Besides the presentation of OSM, the type of interaction and the level of control between the student and the model can be different in models and it is closely related to metacognition (Bull & Kay, 2008).

Most OSM only provide the most basic level of control where the user can only view the model but there is no additional interactivity between the model and the student. These types of models where only the system has full control over the model are called inspectable student models which means the student cannot make or suggest any changes about their models (Bull & Kay, 2010).

On the other hand, editable student models, such as Flexi-OLM, allow the students not only to view their model but also to discuss the contents of their model by providing answer to the question explicitly requested by the system (Mabbott & Bull, 2006). The system and the student can disagree about the content of student model. The system can also offer to show the student why any part of the student model has its current value. The student can also challenge their model so that not only their suggestion would affect their model, but also the information provided by the student can be used as a feedback into the student modelling process.

Co-operative OSM (Beck, Stern, & Woolf, 1997), provides an intermediate level of student control where the student and the system can provide different and complementary information to the student model, and the system only allows the student to change the model if the student can show their self-assessment was accurate (Mabbott & Bull, 2006). This can involve providing evidence and negotiations. The interaction of the system and the student for negotiating the OSM can also be mediated by an artificial agent (such as CALM system) (Kerly & Bull, 2008).

The benefits of allowing the student to update their models are (Bull & Kay, 2010):

- They can inform the system if their knowledge is increased by learning outside the system.
- They can inform the system if they forgot some concepts.

Editable OSM encourages student to take responsibility for their learning interaction and thus promotes meta-cognitive activities (Bull & Kay, 2010).

Since we used EER-Tutor as the test platform for our research, we describe EER-Tutor here.

2.5 EER-Tutor

KERMIT (Suraweera & Mitrovic, 2002) is a stand-alone constraint-based ITS for teaching basic Entity-Relationship (ER) modelling that is developed by the Intelligent Computing Tutoring Group (ICTG) at the University of Canterbury (Figure 14). KERMIT was shown to improve student's performance and was extended to and eventually replaced by EER-Tutor (Martin & Mitrovic, 2002a, 2002b).

With EER-Tutor students can learn Enhanced Entity-Relationship (EER) modelling (Suraweera & Mitrovic, 2004) with a larger set of constraints and problems to cover enhanced database modelling (Zakharov, Mitrovic, & Ohlsson, 2005). EER-Tutor consists of a server side and a client side. The client is in charge of presenting a set of dynamic web pages to the student. The client side of EER-Tutor is designed as HTML pages for end user which are displayable using any web browser.

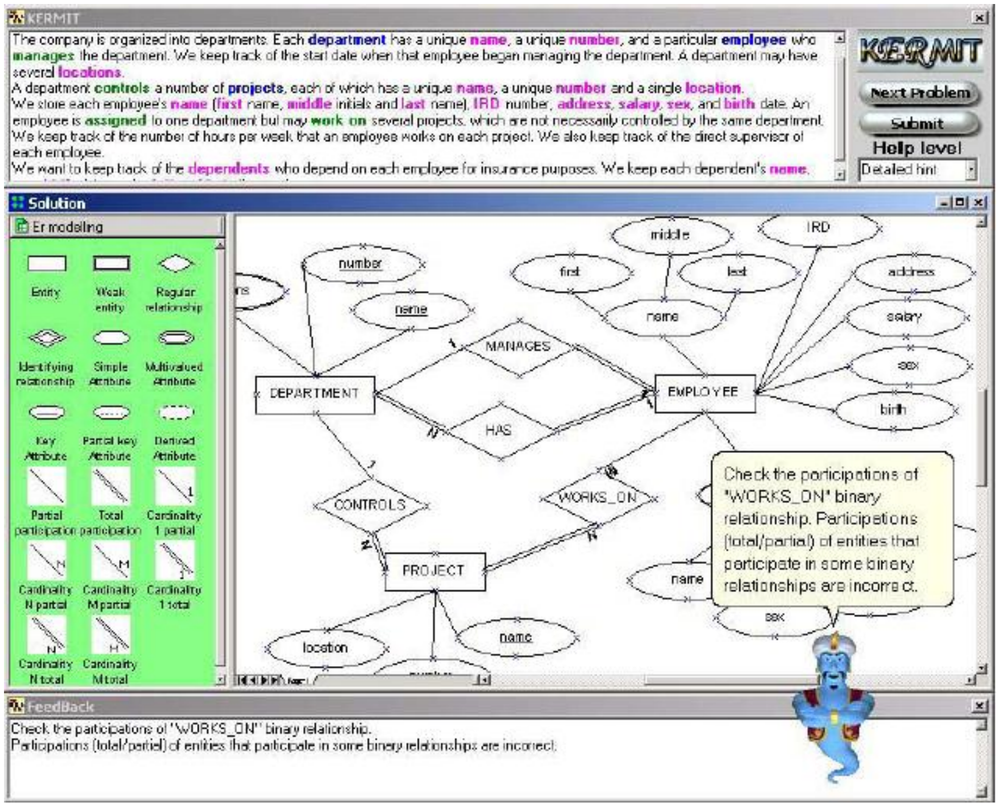


Figure 14: User interface of KERMIT. Adapted from (Suraweera & Mitrovic, 2002).

The server is in charge of processing students' submissions and generating feedback and recording student's interaction with the system. Figure 15 summarizes the architecture of EER-Tutor.

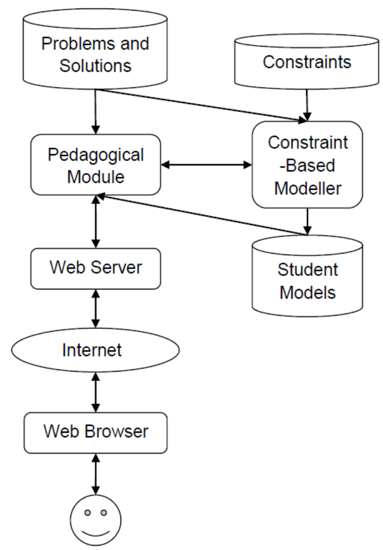


Figure 15: Architecture of EER-Tutor. Adapted from (Duan, 2009).

Figure 16 shows EER-Tutor's workplace page. The center of the page includes a textual description of problem statement and drawing area which allows students to draw EER diagrams, using a given list of tools, as solutions to the problems presented by the system. This page also includes three main frames namely navigation frame, submission frame and feedback frame.

The navigation frame, located at the top of the page, contains problem drop-down box and red buttons which are the navigation tools for the student to other pages such as student model.

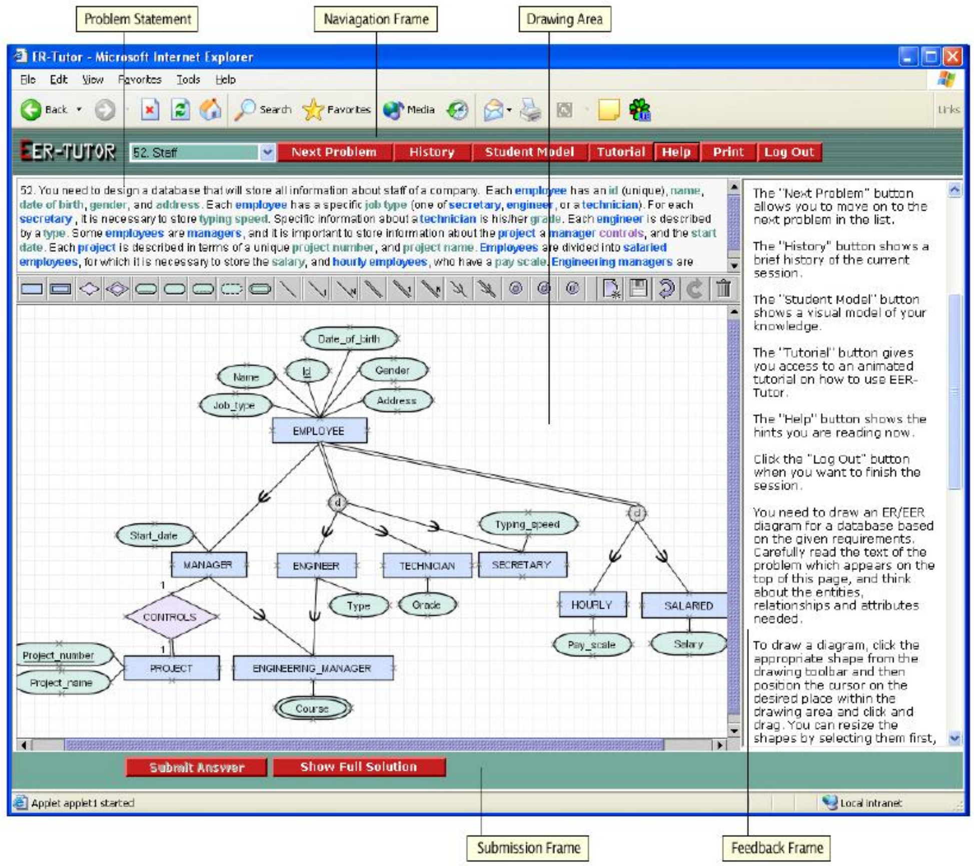


Figure 16: Interface of EER-Tutor. Adapted from (Zakharov et al., 2005).

Submission frame, located at the bottom of the page, contains Submit Answer and Show Full Solution buttons. Once the student finished drawing, he/she can click on them to submit the solution. Feedback will be generated immediately and will be shown inside the Feedback frame, located at right side of the page. In case the student wants to see the ideal solution, he/she can click on Show Full Solution buttons to receive a pop-up page with full solution.

EER-Tutor is a problem solving environment. Once student submits the solutions, pedagogical module sends it to student modeller which diagnoses the solution by comparing it against the predefined ideal solution and syntax and semantic constraints, updates the student model and sends the results of the diagnosis back to the pedagogical module to generate feedback.

2.5.1 Student Model in EER-Tutor

As we mentioned before, EER-Tutor has a constraint-based student model that consists of a short-term and a long-term model (Suraweera & Mitrovic, 2004). The short-term student model records the satisfaction or violation of each relevant constraint for the current problem state. This is done for every solution that is submitted: the student modeller compares the problem state with the ideal solution and the knowledge base. Then it identifies the constraints relevant for the problem state. For relevant constraints, their satisfaction conditions are checked against the problem state. If the satisfaction condition is not met, that is recorded as an error for that solution. The pedagogical module produces individualized feedback per problem based on this short-term model.

EER-Tutor also produces a long-term student model. This long-term student model holds the history of usage of each constraint for the student in the student model log. It records a list of constraint numbers that have been relevant for the student's solutions and whether it has been satisfied or violated each time. The long-term student model is saved when the student signs out and loaded again when the student signs back in.

EER-Tutor has this long-term student model open to students presented in form of skill meters. This model is available to student by clicking on Student Model button in navigation section of the workplace page. This page has a brief introduction of the content of the student model. It also shows the description of colors used in the meters. Figure 17 shows this model. The original EER-Tutor has eight progress bars corresponding to eight concepts in the domain. The length of each bar shows the amount of knowledge in that concept. The green part of a bar demonstrates the correctly learnt amount in a concept but the red part shows the misconception amount; and the white part shows the relative amount of constraints not yet covered. Each bar also has the percentages corresponding to covered and learned knowledge shown in text in front of it.

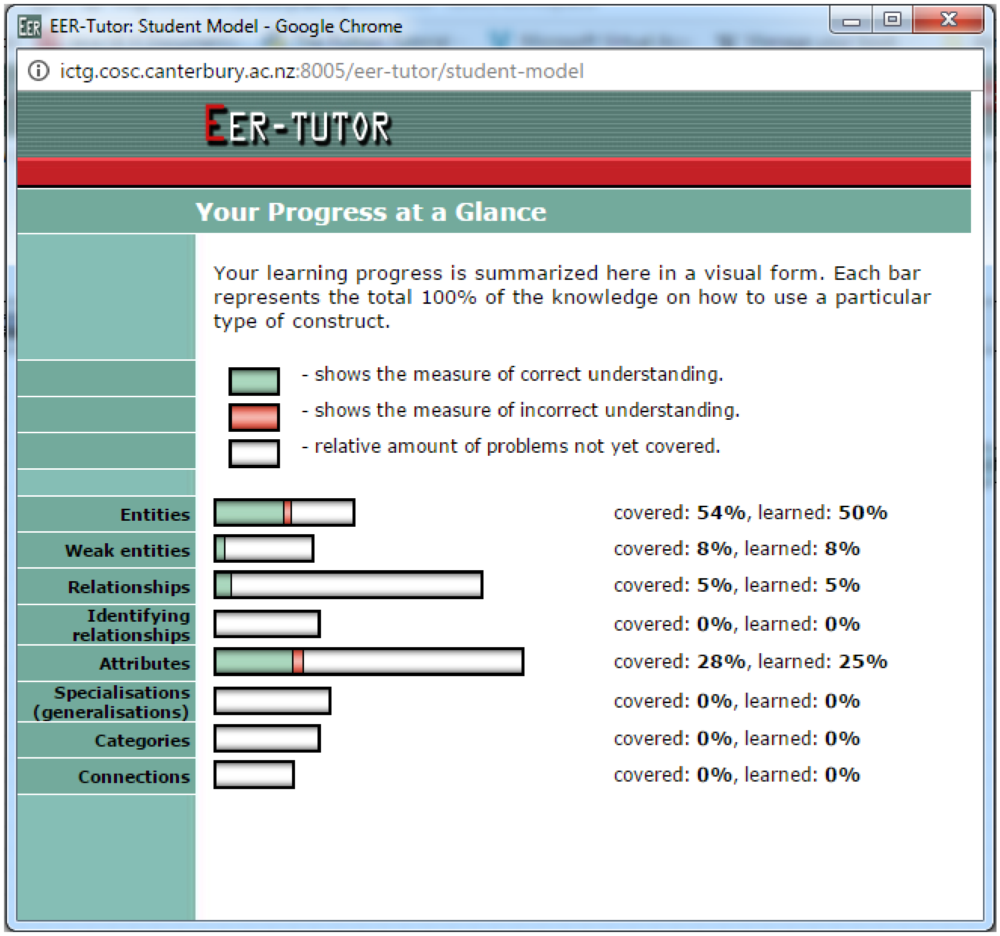


Figure 17: Skill meter in EER-Tutor.

The long-term student model contains information about the student knowledge and progress on problem solving. In the following, I will explain how this progress is calculated in order to be shown to the students.

2.5.2 Progress Bar in EER-Tutor

The student model file is where the EER-Tutor stores the list of constraints used by each user as well as their violation and satisfaction history. The example below is a snapshot of student’s model file stored in the system:

(53 0 (1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1) 4/5)

(50 0 (1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 11 1 1) 4/5)

(49 0 (1 1 1) 1)

(50 0 (1 1 11 1 1 1 1 1 1 1 1 1 1 1 1) 1)

(209 0 (1 1 1 1 1 1) 1)

(16 0 (1 0 0 0 1 0) 1/5)

(208 0 (1 1 1 1 1 1) 1) (14 0 (1 1 0) 2/3)

(64 0 (1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 0 0 0 1 1 1 0) 3/5)

(62 0 (1 1 0) 2/3)

(61 0 (1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1) 1)

...

Each row starts with the constraint number followed by zeros (constraint's violation) and ones (constraint's satisfaction) inside a pair of parentheses which show the usage history of that constraint. Each row also contains the correctness ratio for that constraint at the end, which is the ratio of number of correct attempts in last five attempts by the student.

The formulas below continuously are applied to all the concepts in the OSM:

$$\text{Percentage of coverage} = \frac{Number\ of\ constraints\ relevant\ to\ the\ concept\ in\ the\ student\ model}{Number\ of\ constraints\ relevant\ to\ the\ concept\ in\ knowledge\ base}$$

$$\text{Percentage of correctness} = \frac{sum\ of\ correctness\ ratios\ for\ constraints\ relevent\ to\ the\ concept\ in\ student\ model}{Number\ of\ constraints\ relevant\ to\ the\ concept\ in\ the\ student\ model}$$

The OSM uses these formulas to calculate the relative amount of knowledge for each concept, the amount of taught knowledge so far for each concept and the student's performance on each concept. Then this data is shown to the student on OSM in form of bars and text.

2.6 Cognitive Load Theory

The Cognitive Load Theory (CLT) focuses on the fact that humans have a limited working memory but comparatively unlimited long-term memory and thus it is important for any instructional method to effectively and efficiently use these resources. The working memory has the limited capacity of holding seven plus or minus two chunk of information (Miller, 1956).

CLT discusses that cognitive load has a significant impact on learning of cognitive tasks because the amount of working memory devoted to a task affects how much is learned (Paas & Van Merriënboer, 1994; Sweller, Van Merrienboer, & Paas, 1998). It is shown that the performance can drop because of the under load or overload of learner's cognitive capacity (Young & Stanton, 2002). Thus it is beneficial for an instructional method to properly construct and measure the cognitive load and that is where research in CLT is focused.

In CLT, mental load is defined as the current knowledge about the task or in other words a priori estimate of the cognitive load whereas mental effort reflects the actual cognitive load and is measured while students are working on a task (Paas & Van Merriënboer, 1994). Performance in CLT is referred to learner's achievement while working on a task for example the number of correct answers or number of errors (Paas & Van Merriënboer, 1994).

Cognitive load can be assessed by measuring mental load, mental effort and performance. Analytical and empirical methods are used to measure cognitive load. Rating scales are one of the techniques used for measuring cognitive load where self-rating questions for learners are used. It has been shown that people are capable of giving a numerical value for their mental burden (Linton, Plamondon, Dick, Bittner Jr, & Christ, 1989; Xie & Salvendy, 2000).

Paas and Van Merriënboer (1993) showed that the complex relation between mental effort and performance can be a good measure to rank mental efficiency of an instructional method. This means if the performance of the learner is higher than expected based on their invested mental effort or equally if their invested mental effort is lower than expected based on their performance then that instructional condition is considered more efficient.

In other words, learning more efficiently can be indicated by a combination of high performance and low mental effort. This description of efficiency has been used as the measure for researchers and instructional designers to compare the effect of different instructional methods on learning and is formulated as below:

$$E = |\frac{Z_P - Z_{ME}}{\sqrt{2}}|$$

Where Z_{ME} is defined as the normalized score for mental effort and Z_P is defined as the normalized score for performance (Paas & Van Merriënboer, 1993).

The original calculation of efficiency in (Paas & Van Merriënboer, 1993) used the test performance and mental effort spent on the test while some other studies used the mental effort spent during training and the test performance to measure the efficiency. (Camp, Paas, Rikers, & van Merrienboer, 2001) used this second type of measuring efficiency to personalize the problem selection in the instructional system to better match the learner cognitive state.

2.7 Problem Selection and Mastery Learning

Study choice or problem selection is one of the metacognitive skills. Problem selection in some student models means that student gets the choice of picking the material to work on next without any direct tutoring guidance from the ITS. The student should choose the most appropriate problem to work on based on their weaknesses and strengths. The goal of OSM is to help students become better at problem selection so that they can select better problem when there is no ITS support for problem selection.

It is shown that students prefer to have control over their problem selection (Clark & Mayer, 2016) but that often leads to worse learning outcome comparing to system-selected problem conditions (Atkinson, 1972). This fact motivated research on how to improve students' problem selection skills (Long, Aman, & Aleven, 2015). Mitrovic and Martin (2003) showed that in a faded condition where student went form system-selected problems to student-selected problems gradually, novice students learning gains were significantly improved.

Long and Aleven (2013) present a shared control for problem selection between ITS and students for linear equation solving to help with motivating the student in developing problem selection skills and at the same time avoiding decisions that could degrade the learning. Some previous work mainly focused on scaffolding problem selection in learning. But little work in literature focused on how to design the learning environment to help student with making effective problem selection decisions. Problem selection decision is a transferable skill that can be applied when the scaffolding is not in effect. They discussed the effect of showing progress on problem level (they call it MasteryBar) in addition to showing progress in concepts in OSM. They were interested to study if letting the student choose the next problem would help improving students learning outcome since students can apply the result of their assessment. In their study they used knowledge tracing and mastery learning to adaptively select problems so that students reach mastery on all skills (we will describe the theory of mastery learning in the following). That means every time students want to select a new problem, students can get one problem from a level (complexity level) only if that level is not fully learnt (mastered). Figure 18 shows the view related to their problem selection. In this study, student must master all levels to complete the tutor. Their result showed that their proposed problem selection strategy did not have a significant effect on the post-test score but helped students so that they made fewer incorrect attempts and asked for fewer hints. The student also solved fewer problems to reach mastery. None of these effects were significant though and only the average assistance score was improved significantly. Thus more studies are required to investigate how to design an effective problem selection technique that enhances learning outcome.

Figure 18: OSM in a Linear Equation Tutor. Adapted from (Long et al., 2015).

The work in (Long et al., 2015) is the follow-up on previous work in (Long & Aleven, 2013) again focusing on designing an ITS for equation solving where it provides the student with the full control over problem selection that can motivate students to learn how to develop effective problem selection strategies. This design also presented mastery learning, displayed by MasteryBar, in each level of problems on the problem selection view, but student could select problems from a level even after they mastered that level. It is shown previously that a system controlled problem selection which follows the rule of mastery learning can improve student's learning significantly (Corbett & Anderson, 1992). This work (Long et al., 2015) focuses on examining the impact of student controlled problem selection following the mastery learning on problem selection skills. They took a user–centered technique for their design by running experiments and interviews to investigate how students select problems and how well the students understand the mastery learning and how motivated they are to follow it. Long and Aleven showed their problem selection technique helps students to choose fewer problems from mastered level. They also found out that students normally do not challenge themselves to attempt more difficult levels and did not fully understand the mastery rule thus it should be taught to the students explicitly. Therefore, they adapted their design to include a tutorial on mastery learning and also provide feedback messages to remind and encourage them to work on unmastered levels.

Long and Aleven also discussed motivation has a significant effect on applying the problem selection strategies effectively and found out students normally are not very motivated to exercise higher complexity problems. Thus they adapted their design to include daily challenges and achievements to improve student's motivation to work on higher level unmastered problems.

2.8 Theory of Mastery Learning

Bloom (1968) presented the Theory of Mastery Learning which describes the differences between the conventional model of learning in school and mastery learning.

In the conventional school model, the initial aptitude of students in a class has no impact on the amount of instruction that they receive since they all receive the same instruction. But since the aptitude for learning varies for students in a class, the uniformly distributed instruction results in differences in final learning result. This means since the aptitude among the students of a class at the beginning of the class is normally distributed, the final score at the end of the class is also normally distributed (Bloom, 1968).

On the other hand, the theory of mastery learning suggests that each student should receive the amount and type of instruction based on their individual needs. This means the instruction should not be uniformly distributed, instead it should vary according to need of student. This type of instruction is shown to lead to uniformly distributed and high performance result at the end of the class. Bloom provides some numbers to quantify the prediction of the effect of mastery rule. For example, following the theory of mastery learning in class results in 90% of students reaching the level of final score previously reached by only top 10% of students.

This means the majority of the class would benefit vastly from this model. It also states that novice students would benefit from this model more than advance students. It also states that student with a weaker background would only need to put more time into leaning in initial stages, and as they master the fundamental material the need for the extra time will disappear and eventually all the students in the class would learn at the same quick pace. Also the need for extra instruction of novice student would be insignificant from the instructional need of advance students over time. Many reviews and evaluation of this model are available in literature (Kulik, Kulik, & Bangert-Drowns, 1990) which show significant effect size of this model.

3 Research Hypotheses

In this chapter we start with research motivation and related work then we explain our research objectives and in the end we present our research hypotheses.

3.1 Research Motivation and Related Work

The most commonly known meta-cognitive skills are goal setting, self-assessment, help seeking, self-monitoring and problem selection. Better self-assessment can lead to better problem selection which can result in more efficient learning (Zimmerman, 2008). Many studies have shown the effect of OSM on improving the meta-cognitive skills (Brusilovsky, Sosnovsky, & Shcherbinina, 2004; Bull, Dimitrova, & McCalla, 2007; Hartley & Mitrovic, 2002). More empirical studies are required to evaluate how to design an effective OSM which can enhance student's metacognition skills such as self-assessment and problem selection (Bull, Dimitrova, et al., 2007). This motivated our research to investigate the effect of information provided in OSM on self-assessment and problem selection in a problem solving ITS.

Long and Aleven (2013) investigated the effect of an OSM and problem selection technique (based on mastery learning) for a linear equation tutor in a 2x2 experiment. They found that their OSM resulted in significant learning gain improvement. They also showed students with access to OSM needed significantly less assistance and made significantly less incorrect attempts. But they found no significant improvement in post-test results only because of their problem selection technique, suggesting that more studies are still needed to investigate how problem selection can improve student's learning outcome.

Another open question in the area of ITS design is motivational design meaning that how to design the ITS to help student to want to use the meta-cognitive skills such as problem selection. In this research, we study the effect of presenting mastery learning on enhancing the student's problem selection skills as well as motivation to solve more challenging problems. On top of that, we study the effect of the information provided to students about the domain knowledge (in terms of number of concepts in their OSM) on self-assessment, and learning outcome.

3.2 Research Objectives

As mentioned in Section **Error! Reference source not found.**, in some OSM, the learner has full control over their learning decisions thus the core function of their OSM is to help student develop meta-cognitive skills such as self-assessment and problem selection skills in order to learn effectively and efficiently. Previous research has shown that these skills can be taught (Bielaczyc et al., 1995). Thus it is critical to study how the visualization and the type of information provided in OSM can impact the development of these meta-cognitive skills. The cognitive load theory indicates that our cognitive system can only process 7± 2 items at any time (Miller, 1956). This puts a limit on the amount of information/items that should be demonstrated to the learner in an efficient OSM design. It is our interest to study if the detailed information provided to the learner can reduce their cognitive load and in return can enhance their metacognition skills. In other words, we are interested to know if this information would encourage them to understand their OSM better and help them improve their problem selection skills such as challenging themselves to work on more problem and attempting complex problems.

The result of research presented in (Mitrovic & Martin, 2007) has revealed that even simple OSMs can significantly improve students' meta-cognitive skills and learning. It is believed that novice students are worse at self-assessment and the study with SQL-Tutor performed in 2000 confirmed that (Mitrovic, 2001). However, with support of even a simple OSM, these students performed significantly better compared to peers of similar abilities without access to OSMs (Mitrovic et al., 2002). The same study also showed that advanced students with access to OSM had additional motivation to spend more time on problem solving. In the next study on SQL-Tutor, Mitrovic and Martin (2003) also found that open student models can help students by reflecting more on their knowledge in order to select problems better. However, all of these studies were on SQL-Tutor and there is no study on EER-Tutor focusing on students' meta-cognitive skills. Also, there was no previous study on the effect of OSM on the level of complexity of problem attempted by students in EER-Tutor.

The OSM of current EER-Tutor only presents the percentages of knowledge state (correct and incorrect) on each concept, thus students can reflect on their learning processes. When students recognize where their weaknesses are, an effective problem selection strategy could be to solve problems particularly relating to those areas. Also when students recognize where their strengths are, they may want to stop solving problems from those concepts and move on to other (possibly more complex) concepts. To allow students select problems themselves, we are interested to map domain concepts to problems. In other words, we would display problems categorized based on the concept they cover. We hypothesize this would help students practice or avoid tasks of certain concepts in the domain based on their self-assessment of their knowledge and hence improve corresponding knowledge levels. This also motivates them to exercise their meta-cognitive skills.

Thus we designed a new strategy for presenting information about the student progress and two main objectives of our research are:

1- To investigate how two different OSM visualizations (Skillmetter and Concept list) impact the students' self-assessment and learning outcomes.
2- To investigate how mastery learning in each problem category affects problem selection skills and student motivation to challenge themselves with higher complexity problems.

We used EER-Tutor (describe in Section 2.5) as our test bed.

In order to evaluate the objectives of this research, first we needed to design two different conditions for our study. Therefore, we designed two comparable conditions: the control and experimental groups. The amount of information provided in their OSM is different for these two groups. In the control group, we used the Skill meter model, which has less information and fewer concepts. In the experimental group, we used the Concept list model, which has more information and more detailed concepts. Also, experimental group can see the mastery learning in their problem selection page which shows the percentage of correctly solved problem in each level of complexity.

For each hypothesis, we develop a measure to quantify the hypothesis and then we compare the measures statistically between the control and experimental group to qualify the hypothesis.

3.3 Research Hypotheses

There are four main hypotheses that correspond to our research objectives. The first two hypotheses are mainly focused on how information provided in OSM impacts student's understanding of their OSM, while the other two hypotheses are focused on problem selection skills:

- Students in the experimental group would benefit more from the OSM and achieve higher learning compared to the control group.
- The experimental group students would be more accurate in self-assessment.
- The experimental group students would be better in problem selection.
- The experimental group students would answer more problems and would challenge themselves to answer more complex problems.

4 Design and Implementation

This chapter describes our design and it consists of two parts. The first section explains our OSM and the second details our new problem selection page.

As our research focuses on the analysis of the modification of existing EER-Tutor OSMs, it is necessary here to describe the existing design as well as our modification before discussing the results of our investigation. We also describe the design of existing problem selection and our proposed problem selection page. We will explicitly describe the differences between the design of control and experimental group and the self-assessment questions provided to students in order to conduct our experiment.

4.1 Open Student Model

In the original EER-Tutor (Figure 19), the student model is accessible to students on demand under the student model button. When the student clicks on the student model button, the OSM is shown to the student in a new pop-up window.

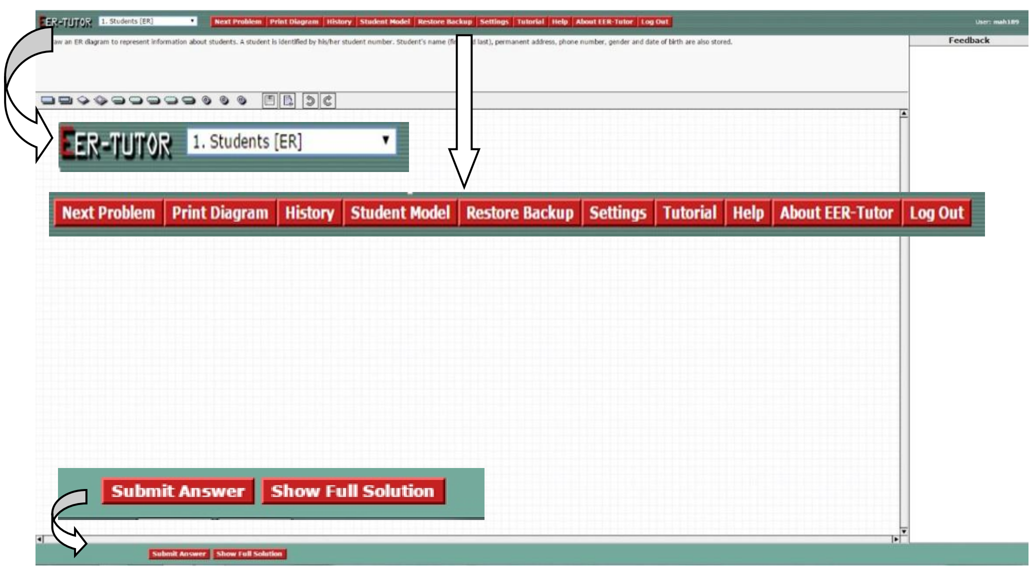


Figure 19: Original EER-Tutor workplace page.

Since the effects of OSM design on metacognition skills are the core objective of this study, we changed the original EER-Tutor so that the OSM is shown to students every time they request a new problem. We changed the OSM page to a normal window instead of pop-up window. This means reviewing OSM is mandatory for the learner before they can choose a new problem. We also changed (removed some of the buttons from) the original EER-Tutor workplace page. Figure 20 shows the new workplace page.

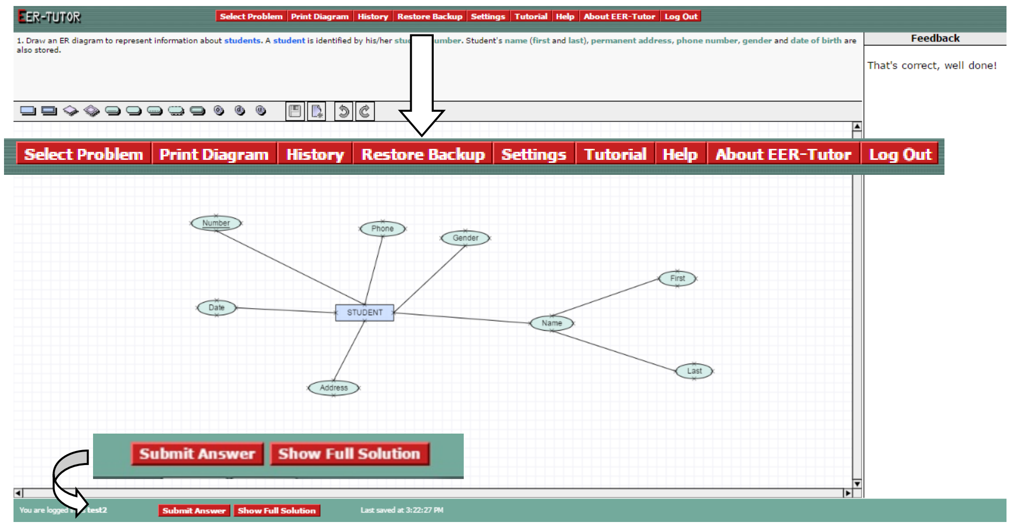


Figure 20: Modified/ new workplace page.

In Chapter 2, we described two common visualizations of OSM called the Skill meter and the Concept list. The original EER-Tutor used the Skill meter as the OSM and the Concept list was designed as an extension of the Skill meter. Figure 17 shows the original Skill meter, and **Error! Reference source not found.** shows the original Concept list in EER-Tutor. The Skill meter model visualizes students' knowledge state divided into eight concepts, while the Concept List contains students' knowledge state in nineteen concepts.

In the control group, we used the Skill meter model and in the experimental group we used the Concept list model as our OSMs. As the core objective of this study is to investigate the effect of different amounts of the information that students received in their

OSM, we chose the Skill meter and the Concept list as OSMs because they are similar to each other and the only difference between these visualizations is the amount of information in terms of number of concepts that they present to the student.

One major step in our design was to decide how the progress bars will be shown to students in order to enhance their self-assessment skills. The idea was to encourage students to compare their progress on each concept before and after the last problem they worked on. In our design, we decided to have two views of the OSM and to put them side by side in a new OSM page so that the student could compare their progress on each concept much more easily. The progress bars in the first view of OSM represent the state of student's knowledge before he/she worked on the last problem, and the progress bars in the second view of the OSM represent student's new knowledge after he/she worked on the last problem. We initially showed the first view of the OSM to the student and after the student pressed the "continue" button in the first view (Figure 21), we showed both views to the student side by side (Figure 22). Thus, the student now could easily identify their progress corresponding to the last problem on each concept by comparing these two views.

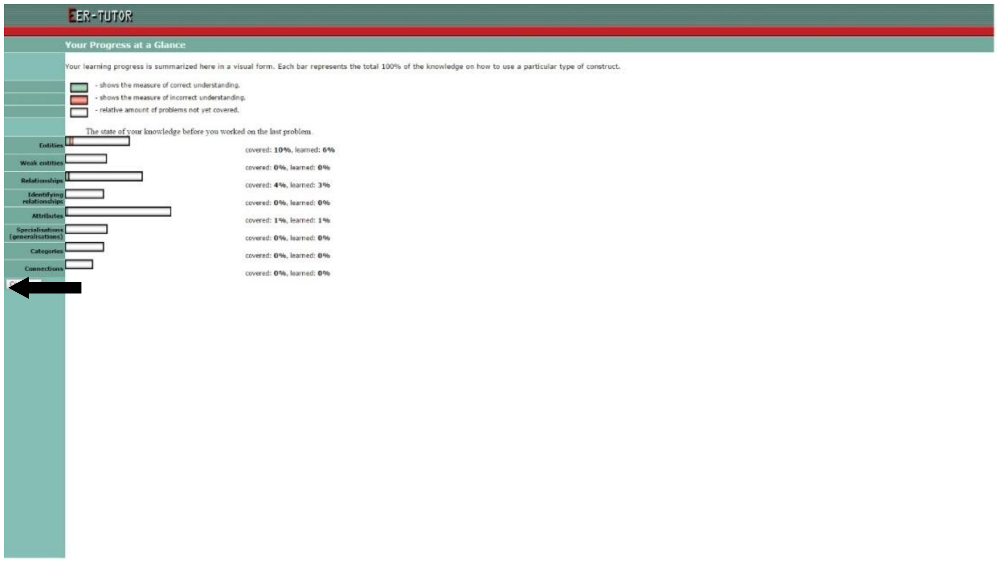


Figure 21: First view ("The state of your knowledge before you worked on the last problem") for the Control group.

The innovation of our proposed presentation shown in OSM is therefore to show the student's progress on each concept before and after the last problem. The progress in these two views is shown as:

- "The state of your knowledge before you worked on the last problem" which is loaded with last knowledge state on all concepts bars.
- "Your New Progress" which is loaded with the new knowledge state on all concepts bars.

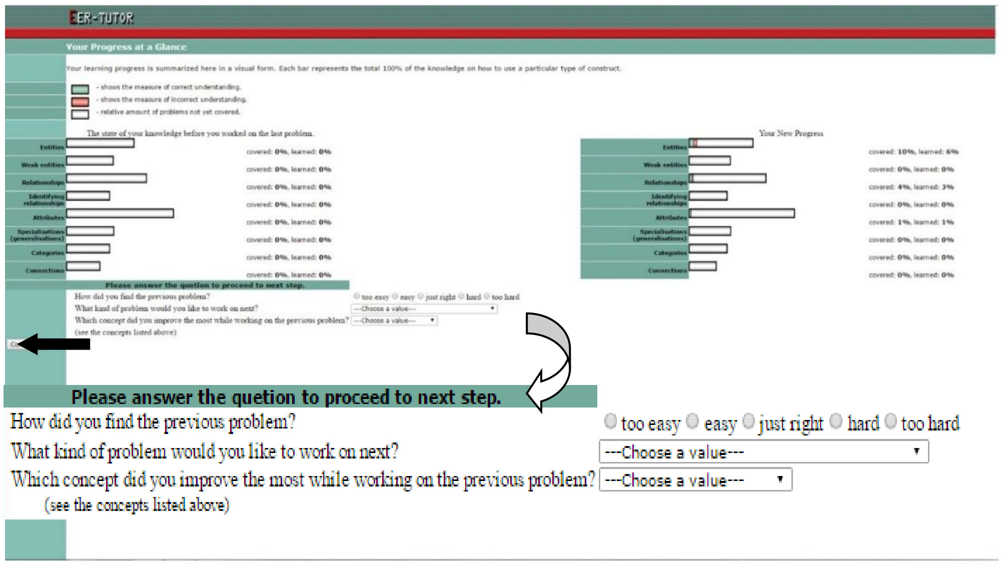


Figure 22: Two views ("The state of your knowledge before you worked on the last problem" and "Your New Progress") for the Control group.

Note that this means the second page includes two views of the Skill meter for the Control group (Figure 22) and two views of the Concept list for the Experimental group (Figure 23). All concepts of "The state of your knowledge before you worked on the last problem" view for the very first problem that the student chose to work on are empty.

Figure 23: Two views ("The state of your knowledge before you worked on the last problem" and "Your New Progress") for the Experimental group.

We used self-assessment questions where participant can indicate their opinion about their experience with the OSM. Each of these questions is corresponding to one of our research hypothesis. Thus they are mainly focused on self-assessment and problem selection. More details about the relation of these questions and the hypotheses are described in Chapter 5.

Every time the student requested a new problem, three self-assessment questions are asked. We decided to add those questions at the end on the OSM page since the OSM is also shown to students every time they request a new problem. The following three self-assessment questions are presented to the student:

1. How did you find the previous problem?

   1-too easy 2-easy 3-just right 4-hard 5-too hard

2. What kind of problem would you like to work on next?

   1- easier than the previous problem.

   2- similar complexity to the previous problem.

   3- harder than the previous problem.

3. Which concept did you improve the most while working on the previous problem?
   Drop down list (list of concepts existing on their OSM)

The student is required to answer all three self-assessment questions before they can proceed to the problem selection page by clicking on continue button (Figure 22 / Figure 23). To prevent students from not answering these questions, we implemented a pop-up alert if they do not answer any of those questions (Figure 24). We collect the answers to these questions with respect to what problem they worked on previously and store them in the student log.

The self-assessment questions can also implicitly help students reflect on their self-assessment and their learning status after each problem. In other words, provided that students inspect the side by side views of their progress before and after each problem, they receive implicit feedback on their self-assessment by answering these questions. Thus these self-assessment questions can prompt the student to review their misunderstandings and strengths and use that information toward selecting the next problem efficiently. It can also keep them stay focused and aware of their problem selection and even motivate them to challenge themselves with attempting complex problems.

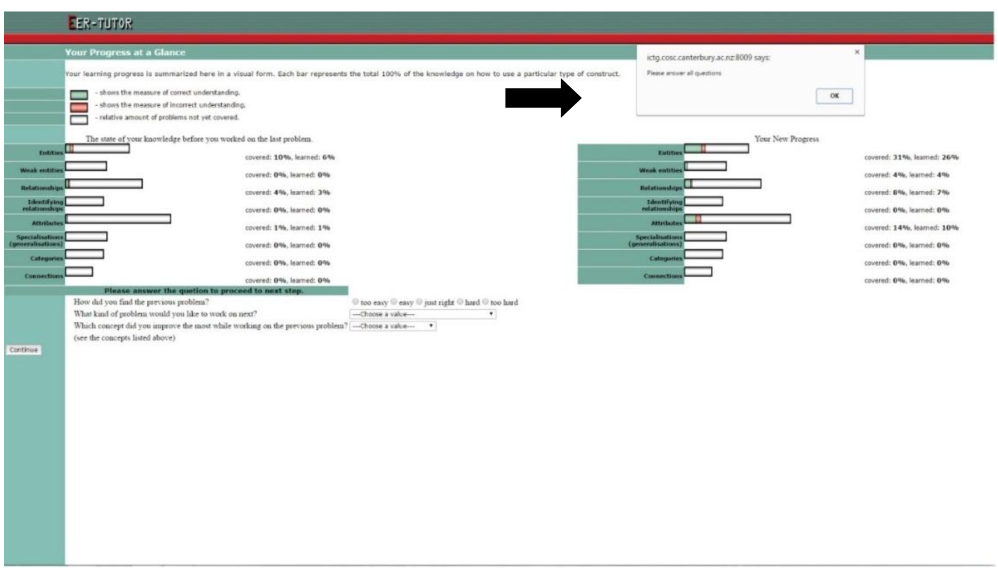


Figure 24: Pop-up alert (top right).

After the student answers these questions, he/she will be directed to the problem selection page. The student can then select a problem from any of the five levels of complexity to work on next. The problem selection page categorizes the problems into five levels, based on their complexity and concepts they cover.

4.2 Problem Selection

In the original EER-Tutor, all problems are viewable to the student under the drop-down list in the workplace page and the problems are not categorized (Figure 19). The student can select any problem to work on next from this drop-down list.

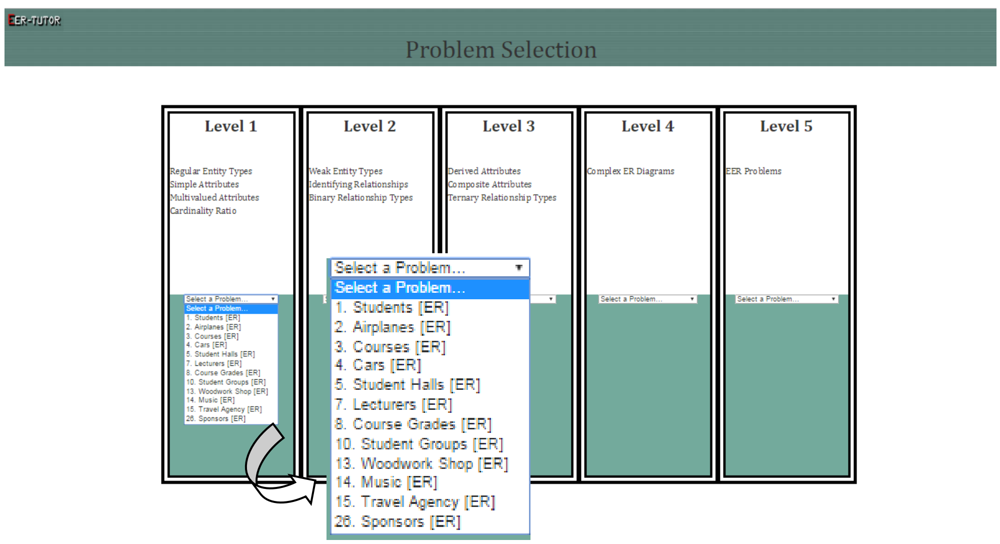


Figure 25: Problem selection for the control group.

In our new design, students are able to select their problem from the new problem selection page after answering self-assessment questions. The idea in our design is to categorize the problems into five levels, based on the complexities of the concepts they cover. Level one represents simpler problems than level two, and so on up to level five. The problems on level five are all of the Enhanced Entity Relationship type.

Each level is described in terms of the domain concepts that they cover:

- Level one: Regular entity types, Simple attributes, Multivalued attributes, Cardinality ratio
- Level two: Weak entity types, Identifying relationship, Binary relationship types
- Level three: Derived attributes, Composite attributes, Ternary relationship types
- Level four: Complex ER problems
- Level five: Enhanced Entity Relationships problems

We divided the 57 problems into five levels: in level one there are 12 problems, level two has 10 problems, level three has 13 problems, level four has 15 problems and level five has 7 problems. Each problem gets a complexity score which is equivalent to the level it belongs to. The new problem selection page is shown in Figure 25 for control group and Figure 26 for experimental group.

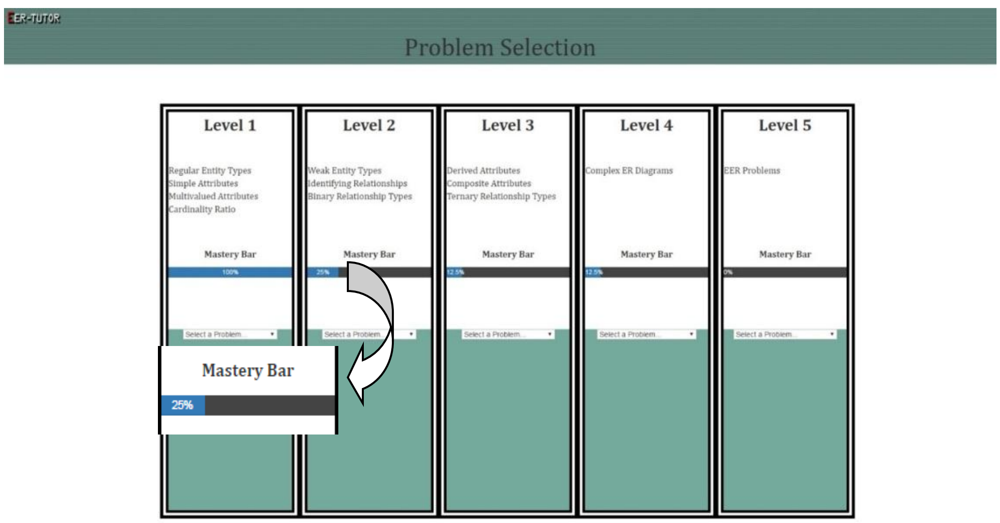


Figure 26: Problem selection for the experimental group.

The goal is to help the student to select appropriate problem to work on. We simply used the complexity score for each level to filter the problems available in that level. We then show the problem id and problem title (see example below) of all problems belonging to that level in a drop down list.

The following code shows an example of how a problem is defined in the system:

```
(1      ; problem id
   1   ; complexity score
     "1. 1. Draw an ER diagram to represent information about <E1> students </E1>.
     A <E1> student </E1> is identified by his/her <E1K1> student number </E1K1>.
     Student's <E1S1> name </E1S1> (<E1S11> first </E1S11> and <E1S12> last
     </E1S12>), <E1S2> permanent address </E1S2>, <E1S4> phone number
     </E1S4>, <E1S3> gender </E1S3> and <E1S5> date of birth </E1S5> are also
     stored."
     (("ENTITIES" "@ E1 STUDENT regular")
      ("RELATIONSHIPS" "")
     ("ATTRIBUTES" "@ E1S1 Name simple composite E1 @ E1S11 First simple
     component E1S1 @ E1S12 Last simple component E1S1 @ E1S2 Address
     simple simple E1 @ E1S3 Gender simple simple E1 @ E1S4 Phone simple
     simple E1 @ E1S5 Dob simple simple E1 @ E1K1 Number key simple E1")
     ("CONNECTIONS" "")
     ("SUPERCLASSES" "")
     ("SUBCLASSES" "")
     ("CATEGORY-SUPERCLASSES" "")
     ("CATEGORIES" ""))
     "1.jpg"
     "Students [ER]" ; problem title
)
```

The number in the first row indicates the problem id and the number in the next row represents the problem complexity score. The text of the problem is presented next, followed by a representation of the ideal solution for the problem and the problem title.

Since one of the objectives of this study is to investigate the effect of mastery progress per level on the students’ problem selection strategies and motivation, the new problem selection page is designed slightly different for control and experimental group. The only difference between the two groups in the problem selection page is that the experimental group can see a "MasteryBar" for each level.

A MasteryBar is shown in Figure 26. The student in the experimental group has five MasteryBars in their problem selection page, which means each complexity level has an individual MasteryBar and these bars represent the mastery of each level as the percentage of correctly solved problems in that level. The student is not required to solve all problems on each level to reach the full mastery. This is because every problem has a different fractional weight.

To calculate the MasteryBar for each level, we need to add the weight of solved problems in that level. We divide the problems in each level into multiple sublevels and each sublevel has a maximum number of solved problems to reach mastery. If we define the number of sublevels by M and the number of problems in a sublevel by N, the maximum number in that sublevel to reach mastery is defined by $\left[\frac{\mathrm{N}}{2}\right]$ and the weight of each problem in that sub-level is defined by $\frac{1}{\mathrm{M}\left[\frac{\mathrm{N}}{2}\right]}$ ([] means the nearest integer function). To calculate the MasteryBar value for each level, we first filter the problems that are solved in each sublevel and then add the weight of solved problems but we stop adding if the total number of solved problems is more than the capacity value of the sublevel. This can be summarized as:

$$\text{sublevel mastery} = \sum_{\mathrm{k}=1}^{\text{sublevel } capacity} \text{weight of solved problem in the sublevel}$$

After calculating this value for each sublevel, we add up the weight of all sublevels belonging to a level to calculate the MasteryBar of that level. The levels were never locked meaning that the student can keep selecting problem from a mastered level as well.

To implement the above design, we created a hash table list which stores the maximum number of each sublevel and weight of each problem as well as the problem ids for each sub-level. We use the value of the complexity score to filter out the problems that belong to that level. For each level, we count the number of problems solved in each sub-level of that level and we use the formula above to calculate the mastery for that sublevel. Then we add up sublevel mastery of all sublevels belonging to that level and display that as the MasteryBar of that level.

4.3 Student Log in EER-Tutor

Student activities are stored by the system in the student log. However, this information is not viewable to the student but is stored in the system to analyse the student behavior and track how they interact with system later. By analyzing the student log, we can track the student during problem solving and can extract necessary information.

Student log stores the student's answers to self-assessment questions as well. Other information stored in the log is the number of student attempts on each problem and if he/she viewed the ideal solution as well.

5 Evaluation

In this chapter, we present the study we ran and the measures we used to quantify our research hypotheses. Then we present the results and our conclusions.

5.1 Classroom Study

The study was conducted with students enrolled in the MBIS623 Data Management course at the University of Canterbury in October 2016. The participants used EER-Tutor in a lab session during the third week of the course. Before the study, the participants had already studied major concepts of the ER domain.

The first time the participants logged into EER-Tutor, they were randomly divided into two groups by the system: the control group and the experimental group. As mentioned before, students in the experimental group had access to the OSM with more detailed concepts and mastery bar. More details about differences between their EER-Tutor are provided in the design and implementation chapter. Students were not aware of the existence of the two groups.

The participants were presented with an information page, which introduced the system. The information page briefly explained the OSM and problem selection page. Since students belonged to different groups, they saw introductions about the OSM corresponding to the group they belong to. After the information page, students proceeded to perform a pre-test online before they could do anything else. The test contained a set of multi-choice questions about data modelling using EER notation. Once students completed the pre-test, they were brought to the main working space, where they could start working on the problems.

The students were given one hour to work with the system excluding pre/post-test. At the end of this period, the participants were asked to do a post-test on paper. The post-test had same number of questions about EER domain and same level of difficulty as the pre-test. There are seven questions in the pre-/post-tests (given in Appendix Pre- and Post-Test). Each question contributes one mark to the maximum score of 7.

There were 27 students who participated in the study and sat both pre-test and post-test (13 students in the control group and 14 in the experimental group). In the rest of this chapter we use the following terms:

- **Solved problem**: referring to a problem that is answered correctly by the student.
- **Attempted problem**: referring to a problem that is attempted at least once. Students can submit answer for a problem as many times as they want before they solve it correctly.
- **Abandoned problem**: referring to an attempted problem which has not been completed.
- **Self-assessment questions**: three questions presented to the student on the OSM page every time the student selects a new problem.

5.2 Hypothesis 1

As we explained in Chapter 3, our first hypothesis is “Students in the experimental group would benefit more from OSM and achieve higher learning outcomes since they have more information about the knowledge domain in their OSMs compared to the control group”.

5.2.1 Comparing Pre/Post-test Scores

The first measure we used for above hypothesis is to compare the pre-test and post-test score of experimental group against control group. Table 1 presents the scores on the pre-test and post-test (means and standard deviations) of experimental and control group. Because our data is not normally distributed, we used the Mann-Whitney U test to compare the two groups. The result shows that there are no significant differences between the pre-test of the two groups (p=0.519). This means these two groups are comparable. The comparison between the pre-test and post-test scores of each group using the Related

Samples Wilcoxon Signed test shows that for each group there is significant improvement in learning (for the control group W=59.5, p = .018, and for the experimental group W=68, p = .022). This means both types of OSM (Skill meter and Concept list) and problem selection have a significant impact on learning outcome of students. On the other hand, the comparison (Mann-Whitney U test) of the post-test scores shows that there is no significant difference in learning outcome of experimental group against control group. This means this hypothesis is not supported. We also calculated the learning gain and normalized gain of each group, but there are no significant differences between the two groups on gains.

Group	Control (13)	Experimental (14)	U, P<.05
Pre-test	2.15 (1.28)	1.86 (1.56)	ns
Post-test	3.85 (1.28)	3.21 (1.25)	ns
Improvement pre- to post-test	W= 59.5, p = .018	W= 68, p = .022	
Learning Gain	1.69 (2.02)	1.36 (1.91)	ns
Normalized Gain	0.279 (0.41)	0.166 (0.45)	ns

Table 1: Summary of pre and post-test average scores (standard deviations reported in parentheses).

We were also planning to compare the less-able and more-able students from the two groups against each other but since the numbers of students in each group were not large enough we could not do that. This is therefore future work to run a bigger study and determine whether less-able students from the experimental group would benefit significantly from OSM comparing to control group.

5.2.2 Comparing Efficiency

As explained in Section 2.6, a well-known measure to compare different instructional methods is defined by the efficiency. We are interested to compare the efficiency of our experimental design against our control design. The instructional Condition Efficiency (E) is defined as below (Paas & Van Merriënboer, 1993):

$$E = |\frac{Z_P - Z_{ME}}{\sqrt{2}}|$$

where Z_{ME} is defined as the normalized score for mental effort and Z_P is defined as the normalized score for performance. This efficiency measure is the mathematical representation of the following statement: if the performance of the learner is higher than expected based on their invested mental effort or equally if their invested mental effort is lower than expected based on their performance then that instructional condition is considered more efficient.

The following question is used for the metric of mental effort:

“How did you find the previous problem? 1-too easy 2-easy 3-just right 4-hard 5-too hard”

The mental effort is defined by the answer to the self-assessment question mentioned above. Note that this question is only answered when student decides to work on a different problem.

If the number of attempts that the student makes to solve a problem is defined by n, then the performance for that question is defined by $\frac{1}{n}$. The higher the number of attempts are, the lower the performance is and that is why we defined the performance by $\frac{1}{n}$. Note that we define the performance to be zero for abandoned problems since the number of attempts made to solve that problem can be assumed to be a very big number.

For each problem we therefore have a mental effort score, ME, which has a value in the range from 1 to 5, and a performance score, P, which has a value of 0 to 1 based on the number of attempts :

We formed two samples for P and two samples for ME as below where μ_i^{ME} is the mean of the mental effort (ME) for problem i and μ_i^{P} is the mean of the performance (P) for problem i:

Experimental Group

$$\begin{bmatrix}\mu_1^{ME}\\ \vdots\\ \mu_i^{ME}\end{bmatrix}\begin{bmatrix}\mu_1^{P}\\ \vdots\\ \mu_i^{P}\end{bmatrix}$$

Control Group

$$\begin{bmatrix}\mu_1^{ME'}\\ \vdots\\ \mu_i^{ME'}\end{bmatrix}\begin{bmatrix}\mu_1^{P'}\\ \vdots\\ \mu_i^{P'}\end{bmatrix}$$

Then we ran a Mann-Whitney U test on these two samples of control against experimental group and we found out that there is no significant difference (p = .135) between means of P and P'. However there is significant difference between mean of ME and ME' (U= 212.5, p = .017). Table 2 presents the means and standard deviations for the two groups.

Group	Control	Experimental	U, P<.05
Performance	.26 (.22)	.36 (.24)	ns
Mental Effort	2.87 (.50)	3.31 (.49)	U= 212.5, p = .017

Table 2: Summary of performance and mental effort scores.

For that reason, in order to evaluate this hypothesis, we computed the means of P over combined students of experimental and control group but we compute the mean of ME of each group separately (since their means were significantly different).

To form Z_P and Z_{ME}, we used the following equations (normalization equations):

η_i^P: the average of performance (combined of both groups) for problem i

η_i^{ME}: the average of mental effort for problem i

δ_i^P: the standard deviation of performance (combined of both groups) for problem i

δ_i^{ME}: the standard deviation of mental effort for problem i

$$Z_i^P = \frac{P_i - \eta_i^P}{\delta_i^P} \qquad Z_i^{ME} = \frac{ME_i - \eta_i^{ME}}{\delta_i^{ME}}$$

$$E_i = \left| \frac{Z_i^P - Z_i^{ME}}{\sqrt{2}} \right|$$

Then we form two samples of efficiency per group as below, E_j represents the mean of efficiency for student j:

Experimental Group $\qquad$ Control Group

$[E_1 \cdots E_j]$ $\qquad$ $[E_1' \cdots E_j']$

We ran the Mann-Whitney U test on these two samples of control against experimental group and we found out that means of efficiency of the two groups are not significantly different (p = .402) with experimental mean= .84 (SD .25) and control mean= 0.73 (SD .24), thus the experimental design is not significantly better than the control design in terms of instructional efficiency. Table 3 summarizes the mean and standard deviation of efficiency for the two groups.

Group	Control (13)	Experimental (14)	U, P<.05
Efficiency	.73 (.24)	.84 (.25)	ns

Table 3: Efficiency.

5.3 Hypothesis 2

As we explained in Chapter 3, our second hypothesis is "The experimental group students would be more accurate in self-assessment since they have more information about the knowledge domain in their OSMs compared to the control group".

We have two questions in self-assessment questions designated for this hypothesis and thus two measures to evaluate this hypothesis. The first measurement is focused on the self-assessment in terms of the level of effort student believes it was required for the problem they solved. The second measure is focused on understanding the OSM and the concept which they think they improved the most while they worked on the last problem.

5.3.1 Self-assessment Accuracy

The following question is designed to be used for the first measure of hypothesis 2: "How did you find the previous problem? 1-too easy 2-easy 3-just right 4-hard 5-too hard."

Therefore b_i is defined as the answer to this question for solved problem i.For each solved problem i, we define m_i and σ_i as the mean and standard deviation of number of attempts made by students to solve this problem. Then for a student, we assign a score for problem i based on the number of attempts made by that student to solve the problem shown in table 4:

Number of attempts made by student to solve problem i	Score assigned to the student for problem i (a_i)
$< m_i - 2 * \sigma_i$	1
$> m_i - 2 * \sigma_i$ and $< m_i - \sigma_i$	2
$> m_i - \sigma_i$ and $< m_i + \sigma_i$	3
$> m_i + \sigma_i$ and $< m_i + 2 * \sigma_i$	4
$> m_i + 2 * \sigma_i$	5

Table 4: Score assignment based on number of attempts student made to solved problem i.

Then we introduce the accuracy index as defined below, where N is the number of problems solved by a student:

$$Accuracy\ index = \frac{1}{N}\sum_{i=1}^{N}(a_i - b_i)^2$$

We formed two samples as below where c_j is the accuracy index for student j:

Experimental Group: $[c_1 \cdots c_j]$

Control Group: $[c'_1 \cdots c'_j]$

Then we ran the Mann-Whitney U test on these two samples and we found out that the means of accuracy are not significantly different (p = .094). Table 5 summarizes the means and standard deviations of the sample we used for this hypothesis.

Group	Control (13)	Experimental (14)	U, P<.05
Accuracy	.52 (.51)	1.25 (1.25)	ns

Table 5: Accuracy.

5.3.2 How well the students understand their OSMs

The following question is designed to be used for the second measure of hypothesis 2. The goal here is mainly to evaluate how well students understand their OSMs (Skill meters for the control group and Concept list for experimental group):

“Which concept did you improve the most while working on the previous problem? Drop down list (list of concepts existing in their OSM)”

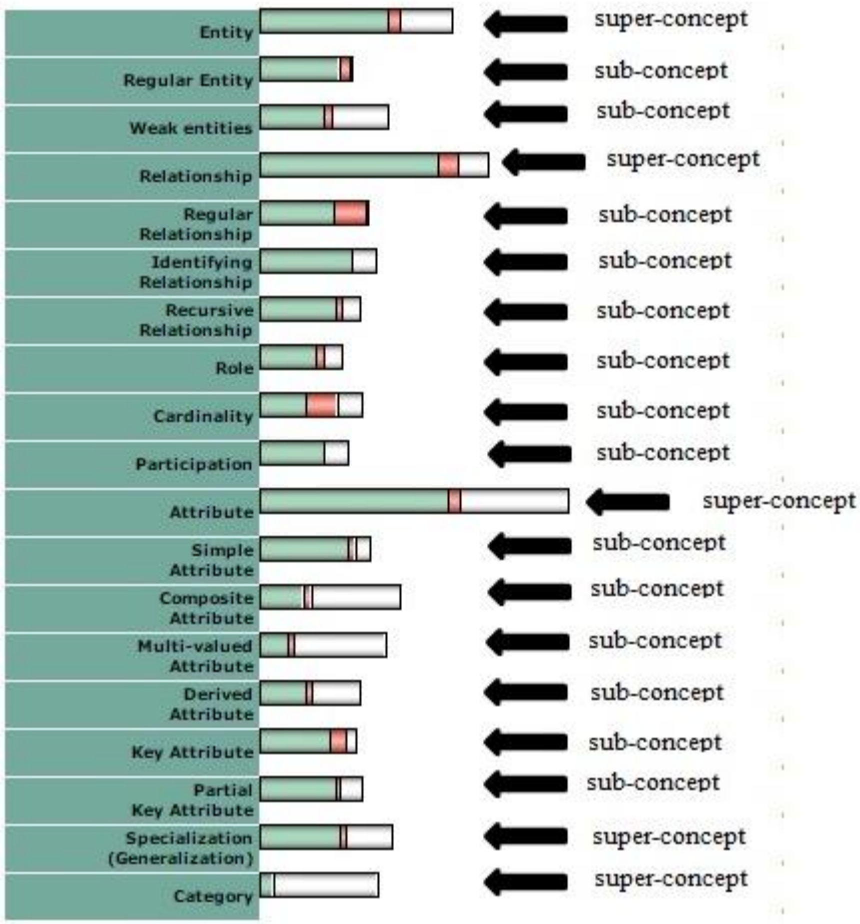


Figure 27: Super-concepts and sub-concepts in the concept list.

The student should be able to compare "The state of your knowledge before you worked on the last problem" view with "Your New Progress" view that we added to the OSM, appearing side by side, and figure out which concept they improved the most. We compare the answers to the question above with the students' actual improvements from their model and assign a matching score for each solved problem as shown in Table 6. The group with higher matching score has a better understanding of their OSM.

As explained before, EER-Tutor calculates and displays the student progress for each concept in the domain in the OSM. Each concept (super-concept) can be divided into multiple sub-concepts. Figure 27 shows super-concepts and related sub-concepts in the concept list as an example.

Most improved concept from Model	Student's answer to most improved concept	Score (M)
Super-concept X	Super-concept X	1
Sub-concept Y	Sub-concept Y	1
Super-concept X	Sub-concept Y of Super-concept X	0.5
Sub-concept Y	Super-concept X of Sub-concept Y	0.5
Sub-concept Y (under Super-concept X)	Sub-concept Z (under Super-concept X)	0.2
Else		0

Table 6: Matching score for all possible conditions of student improvement for concepts.

For each solved problem, we therefore have a matching score, M, which has a value of 0 or 0.2 or 0.5 or 1 based on the matching of their answer to the question above with their model as explained in table 6. We excluded the student answers to this question from our analyzed data if the model shows there is no improvement or negative improvement on all concepts for the solved problem. In total student submitted 190 answers to the self-assessment question above when they proceed to choose the new problems and 80 answers were excluded in this analysis.

We formed two samples as below where α_j is the mean of the matching score for student j:

Experimental Group | Control Group

$$[\alpha_1 \cdots \alpha_j] \qquad [\alpha'_1 \cdots \alpha'_j]$$

Then we ran the Mann-Whitney U test on these two samples and we found out that their means are significantly different (U=23.5, p = .004). Interesting enough the control group has a higher mean (higher score means better matching of model and the answer to the question). Table 7 summarizes the means and standard deviations of the sample we used for this hypothesis.

Group	Control (13)	Experimental (11)	U, P<.05
Matching Score	.56 (0.26)	.26 (0.17)	U=23.5, p = .004

Table 7: Matching Score.

One explanation of this finding is that the control group participants are better at understanding their OSMs because their OSMs are simpler, and contain fewer concepts (from the drop down list) to choose from thus their range of choices is narrower. The control group has fewer concepts in their drop down box than the experimental group and thus the chance of even a random answer to the question above matching their model is higher. This means students generally do not do well in terms of understanding their OSMs. Another explanation could be that sample size of our classroom study was not large enough, thus our future work is to repeat the study with more participants.

Our future work for this hypothesis is to change the answer to the question to include the super-concept and sub-concept to evaluate how accurate the students can match their model. In other word, we would break the drop-down list (Figure 22) to first show only all super-concepts. Then once the super-concept is picked, to show the sub-concepts. Also we should add the "no improvement" answer to the list of answers for this question.

5.4 Hypothesis 3

As we explained in Chapter 3, our third hypothesis is "The experimental group students would be better in problem selection since they have access to mastery learning for each level of problem categories on their problem selection page".

For this hypothesis we used three different measures. The first one is focused on number of attempts on solved problem and the second one is focused on number of solved problems and the third one is focused on number of abandoned problems.

The first measure is to compare the number of attempts for solved problems between experimental and control group. The group with fewer attempts for solved problems would show better problem selection skills. We exclude abandoned problems from this measure because it disturbs the fact that less attempt means better problem selection skills. We only included problems that are solved by either control group or experimental group.

The second measure is to compare the number of solved problems between experimental and control group. The group with more solved problems would show better problem selection skills.

We formed two samples as below where β_i is the mean of attempts to solve problem i and Σ_j is the number of solved problems by student j:

Experimental Group

$$\begin{bmatrix}\beta_1\\ \vdots\\ \beta_i\end{bmatrix}, \begin{bmatrix}\Sigma_1\\ \vdots\\ \Sigma_j\end{bmatrix}$$

Control Group

$$\begin{bmatrix}\beta'_1\\ \vdots\\ \beta'_i\end{bmatrix}, \begin{bmatrix}\Sigma'_1\\ \vdots\\ \Sigma'_j\end{bmatrix}$$

Then we ran the Mann-Whitney U test on the mean of attempts on each problem (β samples) from control against experimental group and we found out that their mean number of attempts are not significantly different (p = .297). Mean of attempts of experimental group is 4.94 (SD 3.12) and is 5.92 (SD 2.74) for control group. This result shows that experimental group made less number of attempts on average on the problems they solved comparing to control group but the mean of the number of attempts is not significantly less than control group.

We also ran the Mann-Whitney U test on the number of problems solved by students (Σ samples) from control against experimental group. We found that their mean is not significantly different (p=0.094). Mean of number of solved problems of experimental group is 6.57 (SD 3.8) and is 4.62 (SD 2.26) for control group. This result shows the experimental group on average solved more problems comparing to control group but the mean of number of problems solved by experimental group is not significantly higher than the control group.

The third measure is to compare the number of abandoned problems between experimental and control group. The group with less number of abandoned problems would be identified with better problem selection skills. We formed two samples (on problems abandoned by either of the groups) as below where π_j is the number of abandoned problems by student j:

Experimental Group

$$\begin{bmatrix} \pi_1 \\ \vdots \\ \pi_j \end{bmatrix}$$

Control Group

$$\begin{bmatrix} \pi'_1 \\ \vdots \\ \pi'_j \end{bmatrix}$$

Then we ran the Mann-Whitney U test on these two samples of control against experimental group and we found out that their mean is not significantly different. Mean of control group is 1.31 (SD 1.55) while mean of experimental group is .93 (.917). This result only shows that experimental group abandoned fewer problems on average but not significantly less than control group. This result shows the experimental group not only solved more problems with fewer attempts but also abandoned fewer problems. Table 8 summarizes the mean and standard of samples we used for this theory.

Group	Control	Experimental	U, p<.05
Attempts on Solved Problems	5.92 (2.74)	4.94 (3.12)	ns
Number of Solved Problems	4.62 (2.26)	6.57 (3.8)	ns
Number of Abandoned Problems	1.31 (1.55)	.93 (.917)	ns

Table 8: Summary of hypothesis 3.

5.5 Hypothesis 4

As we explained in Chapter 3, our fourth hypothesis is "The experimental group students would attempt more difficult problems and would be motivated to challenge themselves to attempt complex problems more than the control group participants".

The first measure of this hypothesis is to compare the number of attempted problems between the two groups for each level of complexity. Since this OSM also has all problems divided to 5 level of complexity, we studied the distribution of attempted problems among the two groups. We define the complexity level of attempted problem j by c_j which can have a value from 1 to 5. Figure 28 shows this distribution. This figure shows both groups did not attempt any level 4 and 5 problems, and they mainly attempted level 1 problems.

The second measure of this hypothesis is to compare the motivation to attempt problems from higher level of complexity between the two groups. The following self-assessment question is designed to be used for the second measure of this hypothesis:

"What kind of problem would you like to work on next? 1-easier than the previous problem 2-similar complexity to the previous problem 3-harder than the previous problem."

We define the answer to this question for the attempted problem j by q_j. We are interested to measure how motivated students are to solve more complex problems. Therefore, we compare the answer to question above between the experimental and control group. Note that if the students' answer to question above is "similar complexity" while they were working on a level one complexity problem, their motivation to attempt higher complexity problem is not the same as if they answer "similar complexity" while they were working on a level three complexity problem. Thus the motivation measure is not only defined by q_j, instead it should be defined by $\mathrm{q_j} * \mathrm{c_j}$.

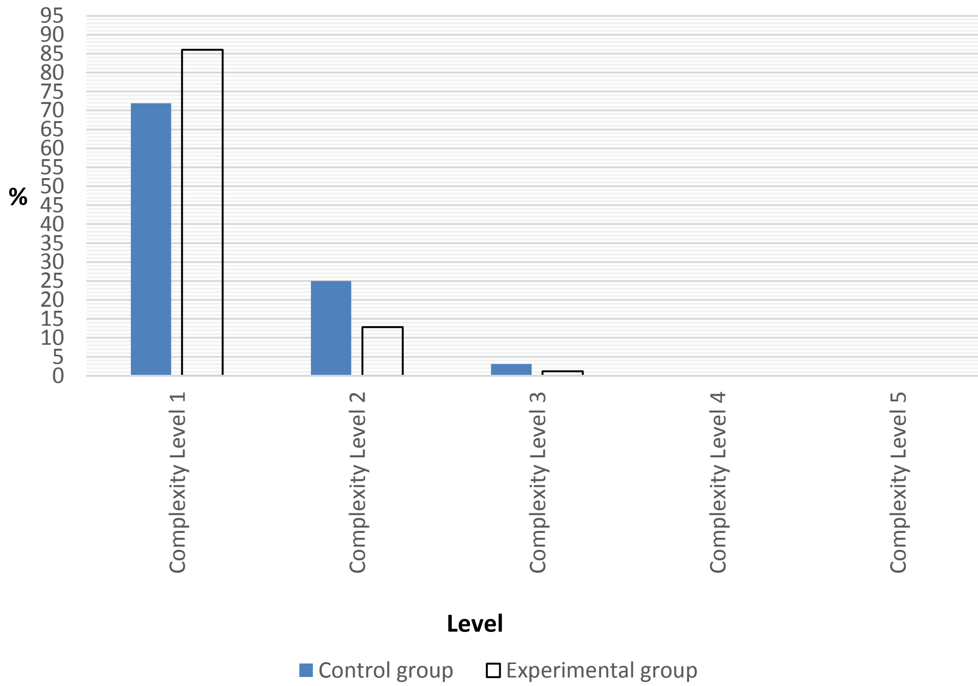


Figure 28: Distribution (Percentage) of attempted problems at different levels.

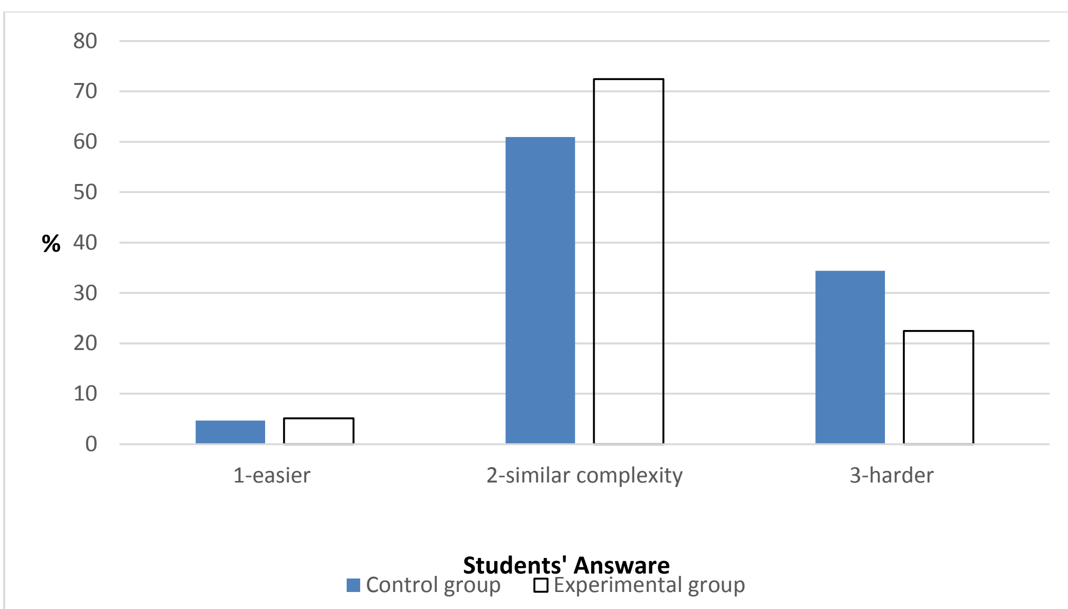


Figure 29: Distribution (percentage) of students' answers to self-assessment question (above question).

Figure 29 shows the distribution of answers for the two groups. Experimental group attempted more problems and thus they answered to this question more. The figure shows both groups are mainly motivated to answer the problems with same complexity.

We also ran some statistical analysis on our classroom data. We formed two samples as below where λ_i and ω_i are the score of student i as explained below:

c_j: complexity level of attempted problem j

q_j: answer to the self-assessment question for the attempted problem j

λ_i: average complexity of attempted problems for student i

ω_i: average complexity of problems student i would like to attempt

N: total number of attempted problems for student i

Experimental Group	Control Group
$[\lambda_1 \ldots \lambda_i], [\omega_1 \ldots \omega_i]$	$[\lambda'_1 \ldots \lambda'_i], [\omega'_1 \ldots \omega'_i]$

$$\lambda_i = \frac{1}{N} \sum_{\text{attempted problem j of student i}} c_j$$

$$\omega_i = \frac{1}{N} \sum_{\text{attempted problem j of student i}} q_j * c_j$$

Then we ran a Mann-Whitney U test on λ and λ'samples of control against experimental group and we found out that their mean is not significantly different ($p = .943$). The mean of experimental group is 1.3 (SD .275) and the mean of control group is 1.4 (SD .512). This result shows that student from control group on average attempted problems with higher complexity (but not significantly higher) comparing to experimental group. Figure 28 also confirms this result and shows that experimental group only solved higher percentage of problem from level one comparing to control group, but control group solved higher percentage of problems in level two and level three.

Then we ran a Mann-Whitney U test on ω and ω'samples of control against experimental group and we found out that their mean is not significantly different (p =

.685). The mean of experimental group is 2.75 (SD .63) and the mean of control group is 3.16 (SD 1.36). This result show that the student from experimental group on average are not motivated to attempt problem with higher complexity comparing to control group but their average is not significantly different. Figure 29 also confirms this result and it shows experimental group on average wants to solve problem with same complexity but not harder comparing to control group.

As explained in design and implementation chapter, experimental group can see their mastery progress on each level of problem complexity in their problem selection page. We are interested to analyse if students in experimental group could understand and follow mastery rule, meaning that they would stop selecting problem from a level they have already fully mastered. Thus we divide problems to two categories:

- **Unmastered problems**: problems (in each level of complexity) that are attempted before that level is 100% mastered.
- **Mastered problems**: problems (in each level of complexity) that are attempted after that level is 100% mastered

Table 9 shows the average percentage of mastered (number of mastered problems/total number of attempted problems) and un-mastered (number of un-mastered problems/total number of attempted problems) problem for each level of complexity. On average, experimental group attempted 13% mastered problem and 87% un-mastered problems, where control group attempted 13% mastered problem and 87% un-mastered problems.

	Un-mastered	Un-mastered	Un-mastered	Total Un-mastered	Mastered
	L1	L2	L3		L1M
Control	.51 (.35)	.3 (.35)	.06 (.12)	.87	.13 (.13)
Experimental	.59 (.2)	.25 (.18)	.04 (.13)	.87	.13 (.1)

Table 9: Percentage of mastered and un-mastered attempted problem on each level of complexity.

The only level that has mastered problem attempted is level one. For level one, we formed two samples as below:

Experimental Group Control Group

$$\begin{bmatrix} l_1 \\ \vdots \\ l_j \end{bmatrix} \qquad \begin{bmatrix} l'_1 \\ \vdots \\ l'_j \end{bmatrix}$$

Where l_j is the percentage of mastered problem attempted by student j. Then we ran a Mann-Whitney U test on l and l'samples of control against experimental group and we found out that their mean is not significantly different (p = .905). The mean of experimental group is .13 (SD .1) and the mean of control group is .13 (SD .13). This result shows that experimental group attempted almost same percentage of problem from a mastered level comparing to control group.

6 Conclusions

The goal of ITS is to mimic a human one-to-one tutoring. There has been a significant amount of research and accomplishment to achieve this goal but ITS is still far from perfectly providing personalized learning support.

OSM is known to help student developing meta-cognitive skills, but little study in literature focused on how to design an efficient and effective OSM. Instructional efficiency is a well-known measure to compare different instructional designs but little research is done how information in OSM would affect this efficiency. This was our motivation to study the effect of our new strategy to present the knowledge state in OSM on instructional efficiency and meta-cognition skills.

In the area of problem selection, most research is focused on scaffolding problem selection support by ITS. Thus we were motivated to study how to improve problem selection skills in a ITS with OSM. Another open question in the area of ITS design is motivational design meaning that how to design the ITS to help student to want to utilize the meta-cognitive skills. This was our motivation to study the effect of our new strategy on encouraging student to learn and use the mastery learning theory toward improving problem selection strategies.

In the following we review the conclusions we draw from our evaluation and address issues and future work that came to our attention based on the results of our evaluation.

Our research mainly focused on two subjects:

1- How does the information presented in the students’ OSM impact their understanding of their OSM? Can OSMs improve their self-assessment and learning?

2- How does the information presented to student via problem selection pages impact their problem selection skills and motivation to challenge themselves to solve more complex problems

In the design and implementation chapter, we explained the design for experimental and control group that can verify our hypotheses. The ICTG team had developed a number of ITSs. We picked EER-Tutor as our test bed because there is no study on EER-Tutor focusing on students' meta-cognitive skills. Also, there was no previous study on the effect of OSM on the level of complexity of problem attempted by student in EER-Tutor. We selected skill meter and concept lists for the control and experimental groups respectively, because they are very similar in presentation and the only difference is the number of domain concepts shown. The original EER-Tutor only showed the progress of student per concept, we also added the progress per problem category (problem complexity) to that. We also asked the student to answer three questions about the last problem they worked on. The intention of these questions was not only to help us evaluate our hypotheses, but also to encourage students to pay attention and understand their OSM better and motivate them to work on higher complexity problems.

There were 27 students who participated in our classroom study, 13 students in the control group and 14 students in the experimental group. The ideal number of participants in each group (control and experimental) is more than 15 students but since we missed the timeline for the original course to use the EER-Tutor, we could not meet this ideal number. Thus we were limited in the number of participants, and we could not compare novice versus advanced students in our evaluation for our measures. The other limitation we faced during the classroom study is that the total time students spent exercising on EER-Tutor was around one hour which was short. Some of the data we gathered from the study also had to be not used for our evaluation which was reported in evaluation chapter separately and discussed with details.

The evaluation chapter included the measures to validate our hypotheses and the results. We will review the results briefly here:

The comparison between the pre-test and post-test scores of each individual group shows that there is significant improvement in learning in both groups. But, the comparison of the post-test scores between experimental and control group shows that there is no significant difference in learning outcome of experimental group against control group.

We learned from our evaluation that students generally do not perform well in self-assessment. Also, showing students more detailed domain concepts did not improve their self-assessment or their learning outcome significantly. Our intention was to compare the less-able and more-able students from the two groups on self-assessment skills, but since the numbers of students in each group were not large enough, we could not do that. This is therefore future work to run a bigger study and compare.

We also learned from our evaluation that showing progress of learning on each level of problem complexity helped student with problem selection skills (Experimental group students solved problems with fewer attempts, abandoned fewer problem and solved more problems), but the improvement was not significant.

Our evaluation also showed that students either did not understand the concept of mastery or were not motivated to take advantage of it since they kept solving problems from the levels they have already mastered. We also observed that showing MasteryBars and asking self-assessment question did not motivate them to solve more challenging problems. Thus our future work is to explicitly explain the theory of mastery learning in the OSM to encourage student to learn and follow it. Another future work is to include more explicit hints to encourage the student to avoid selecting problems from levels they have mastered already and instead attempt problems from new and higher complexity levels. To accomplish that, we are planning to change the problem selection page to include a pop-up hint if the student selects problems from mastered levels. This pop-up hint would include a statement that would inform the student that the level is already mastered and if he/she would like to pick another problem from a different level.

Another thing may be valuable to do in terms of encouraging students to attempt higher complexity level of problems is to provide comparisons among peers in OSM. This can be accomplished by showing the student the highest complexity level of problems that class attempted, or even by providing the user a ranking. This enables students to be aware of where they are in the group and can motivate them to challenge themselves.

7 Appendix Pre- and Post- Test

The following is a copy of the pre-test and post-test that was used in the evaluation study described in Chapter 5.

Pre-test (Online)

The following test will help you assess your knowledge of enhanced entity-relationship modeling before you begin using EER-Tutor. Please read the questions carefully and select the appropriate answers.

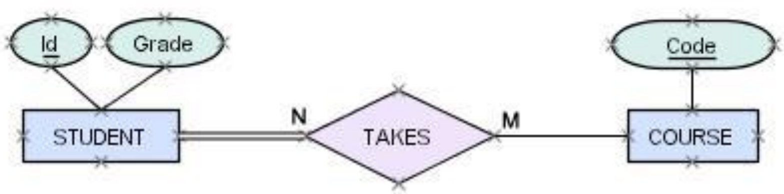
After you submit your solution you'll be able to see correct answers and a detailed explanation of the solutions.

Question 1

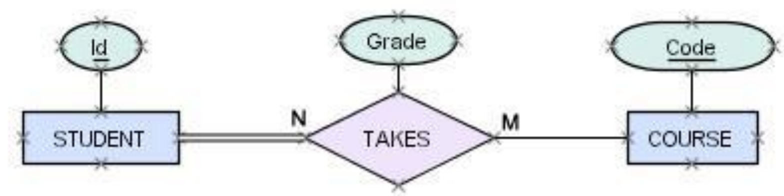
Which of the given ER diagrams corresponds best to the following requirements?

For each course a student has taken, we need to know the final grade. Each course has a unique course code and a student has his/her student id.

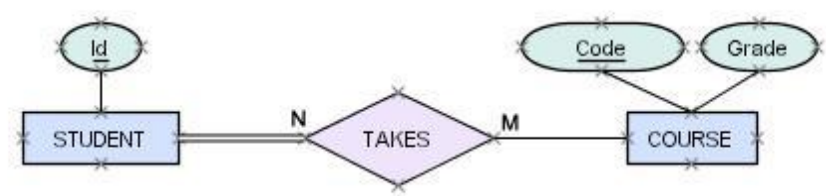
a.



b.



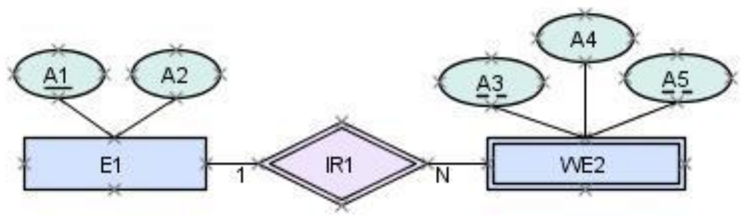
c.



d. Don’t know

Question 2

Is the following diagram correct?



a. Yes.
b. No.
c. Don’t know.

Question 3

The values of the completeness constraint are:

a. Disjoint or partial.

b. Disjoint or overlapping.

c. Partial or total.

d. Overlapping or partial.

e. Don’t know

Question 4

If an entity type has a multi-valued attribute, then

a. Each entity of this type can have one of several values for that attribute.

b. There are some entities of this type that have more that one value for that attribute.

c. Each entity of this type has more than one value for that attribute.

d. There are many valid values for that attribute.

e. Don't know.

Question 5

A weak entity type participates in the identifying relationship type

a. Always totally.

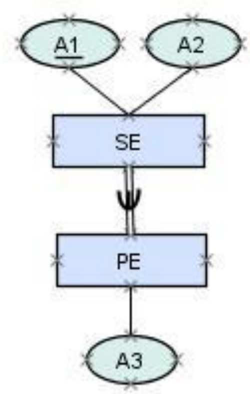
b. Always partially.

c. Either totally or partially.

d. Don't know

Question 6

Is the following EER diagram correct?



a. Yes.

b. No.

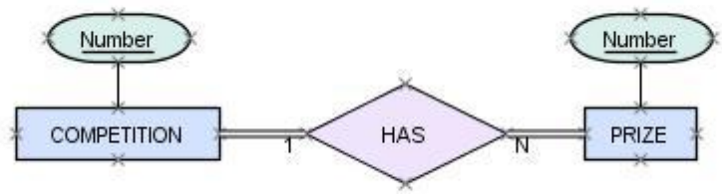
c. Don't know.

Question 7

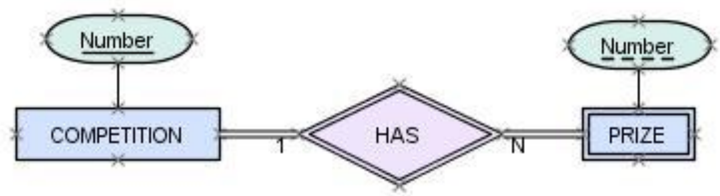
Which of the given ER diagrams corresponds best to the following requirements?

Each competition has a unique number. For every competition there is also a list of prizes identified by numbers unique only within a given list.

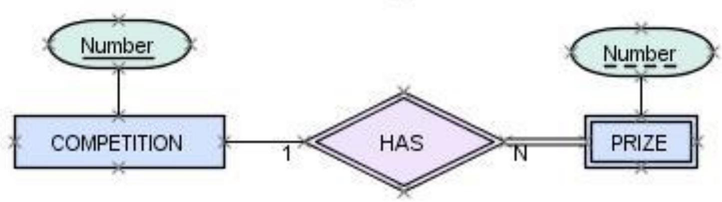
a.



b.



c.



d. Don't know

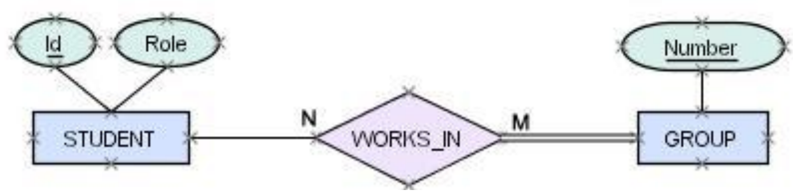
Post-test (in class) **User code:**

Question 1

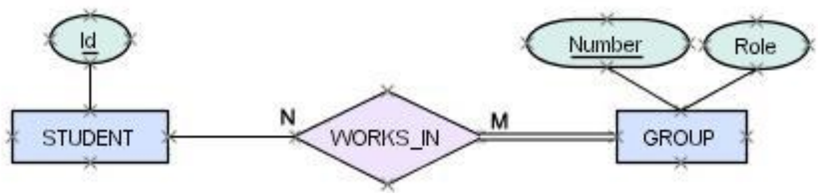
Which of the given ER diagrams corresponds best to the following requirements?

Sometimes students work in groups. A student may be a member of several groups, but he/she may have different roles in different groups. Each student has an id, and each group has a unique number.

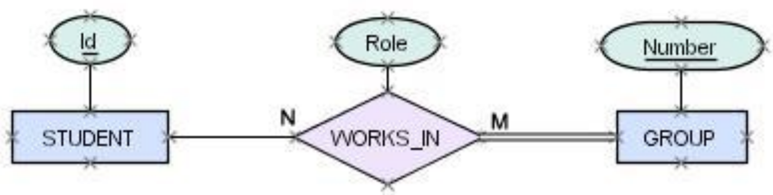
a.



b.



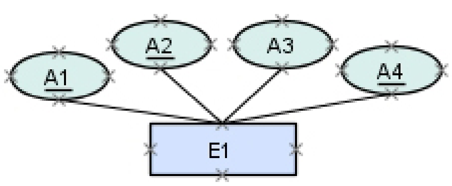
c.



d. Don’t know.

Question 2

Is the following diagram correct?



a. Yes.

b. No.

c. Don’t know.

Question 3

A disjoint specialization means that

a. Every entity belonging to the superclass must belong to exactly one subclass.

b. Every entity from the superclass may belong to one or more subclasses.

c. Some entities from the superclass will appear at the level of subclasses.

d. An entity from the superclass may belong to at most one subclass.

e. Don’t know.

Question 4

A derived attribute is

a. An attribute whose values do not exist for every entity.

b. An attribute that has several components.

c. An attribute whose values are optional.

d. An attribute whose values can be derived from other attributes/relationships.

e. Don’t know.

Question 5

A weak entity type can have

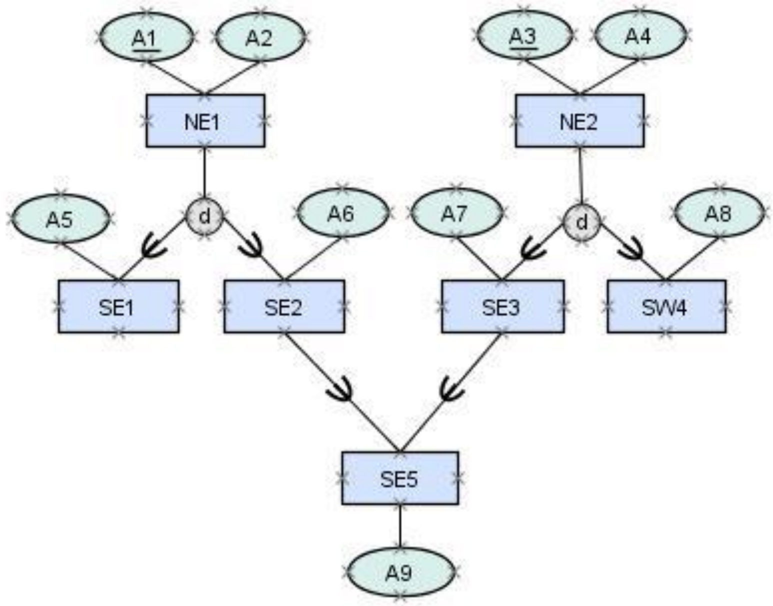
a. Exactly one owner.

b. One or more owners.

c. Don’t know.

Question 6

Is the following diagram correct?



a. Yes.

b. No.

c. Don’t know.

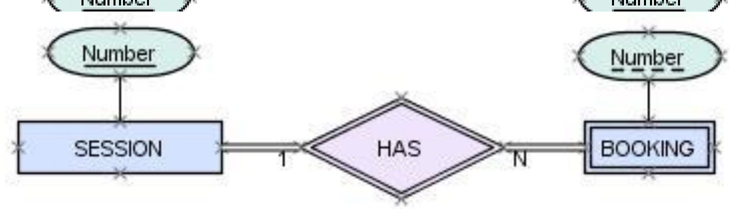
Question 7

Which of the given ER diagrams corresponds best to the following requirements?

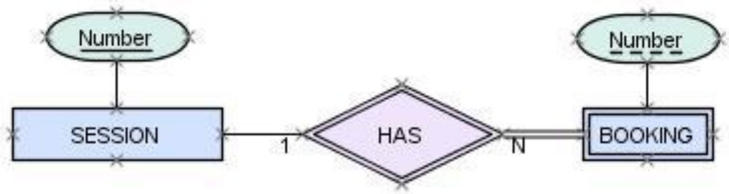
Sessions are identified by unique numbers. For some sessions, there might be lists of bookings identified by numbers unique only within a given list.

a.

b.



c.



d. Don’t know.

8 References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological review, 89*(4), 369.

Anderson, J. R. (Ed.). (1993). *Rules of the mind*: Psychology Press.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The journal of the learning sciences, 4*(2), 167-207.

Anderson, J. R., & Lebiere, C. (Eds.). (1998). *The atomic components of thought Lawrence Erlbaum*.

Atkinson, R. C. (1972). Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology, 96*(1), 124.

Beck, J., Stern, M., & Woolf, B. P. (1997). Cooperative student models. *Artificial Intelligence in Education, IOS Press, Amsterdam*, 127-134.

Bielaczyc, K., Pirolli, P. L., & Brown, A. L. (1995). Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem solving. *Cognition and instruction, 13*(2), 221-252.

Bloom, B. S. (1968). Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1. *Evaluation comment, 1*(2), n2.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher, 13*(6), 4-16.

Brusilovsky, P., Somyürek, S., Guerra, J., Hosseini, R., & Zadorozhny, V. (2015). The value of social: Comparing open student modeling and open social student modeling. *International Conference on User Modeling, Adaptation, and Personalization*, 44-55.

Brusilovsky, P., Sosnovsky, S., & Shcherbinina, O. (2004). QuizGuide: Increasing the educational value of individualized self-assessment quizzes with adaptive navigation support. *Proceedings of World Conference on E-Learning*, 1806-1813.

Bull, S. (2004). Supporting learning with open learner models. *Planning, 29*(14), 1.

Bull, S., Dimitrova, V., & McCalla, G. (2007). Open learner models: Research questions. *International Journal of Artificial Intelligence in Education, 17*(2), 83-87.

Bull, S., & Kay, J. (2007). Student models that invite the learner in: The SMILI:() Open learner modelling framework. *International Journal of Artificial Intelligence in Education, 17*(2), 89-120.

Bull, S., & Kay, J. (2008). Metacognition and open learner models. *The 3rd workshop on meta-cognition and self-regulated learning in educational technologies, at ITS2008*, 7-20.

Bull, S., & Kay, J. (2010). Open learner models *Advances in intelligent tutoring systems* (pp. 301-322): Springer.

Bull, S., Mabbott, A., & Abu Issa, A. S. (2007). UMPTEEN: named and anonymous learner model access for instructors and peers. *International Journal of Artificial Intelligence in Education, 17*(3), 227-253.

Bull, S., & Nghiem, T. (2002). Helping learners to understand themselves with a learner model open to students, peers and instructors. *Proceedings of workshop on individual and group modelling methods that help learners understand themselves, International Conference on Intelligent Tutoring Systems, 2002*, 5-13.

Camp, G., Paas, F., Rikers, R., & van Merrienboer, J. (2001). Dynamic problem selection in air traffic control training: A comparison between performance, mental effort and mental efficiency. *Computers in Human Behavior, 17*(5), 575-595.

Carr, B., & Goldstein, I. P. (Eds.). (1977). *Overlays: A theory of modelling for computer aided instruction*: DTIC Document.

Clark, R. C., & Mayer, R. E. (2016). *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*: John Wiley & Sons.

Corbett, A. T., & Anderson, J. R. (1992). Student modeling and mastery learning in a computer-based programming tutor. *International Conference on Intelligent Tutoring Systems*, 413-420.

Duan, D. (2009). *Evaluating the Effectiveness of Multiple Presentations for Open Student Model in EER-Tutor*.

Fitts, P. M. (1964). Perceptual-motor skill learning. *Categories of human learning, 47*, 381-391.

Freedman, R., Ali, S. S., & McRoy, S. (2000). Links: what is an intelligent tutoring system? *intelligence, 11*(3), 15-16.

Gilmore, D., & Self, J. (1988). The application of machine learning to intelligent tutoring systems. *Artificial Intelligence and Human Learning, Intelligent computer-assisted instruction, New York: Chapman and Hall*, 179-196.

Hartley, D., & Mitrovic, A. (2002). Supporting learning by opening the student model. *International Conference on Intelligent Tutoring Systems*, 453-462.

Holt, P., Dubs, S., Jones, M., & Greer, J. (1994). The state of student modelling *Student modelling: The key to individualized knowledge-based instruction* (pp. 3-35): Springer.

Kay, J. (2008). Lifelong learner modeling for lifelong personalized pervasive learning. *IEEE Transactions on Learning Technologies, 1*(4), 215-228.

Kerly, A., & Bull, S. (2008). Children's interactions with inspectable and negotiated learner models. *International Conference on Intelligent Tutoring Systems*, 132-141.

Kulik, C.-L. C., Kulik, J. A., & Bangert-Drowns, R. L. (1990). Effectiveness of mastery learning programs: A meta-analysis. *Review of educational research, 60*(2), 265-299.

Linton, P. M., Plamondon, B. D., Dick, A., Bittner Jr, A. C., & Christ, R. E. (1989). Operator workload for military system acquisition *Applications of human performance models to system design* (pp. 21-45): Springer.

Long, Y., & Aleven, V. (2013). Supporting students' self-regulated learning with an open learner model in a linear equation tutor. *International Conference on Artificial Intelligence in Education*, 219-228.

Long, Y., Aman, Z., & Aleven, V. (2015). Motivational design in an intelligent tutoring system that helps students make good task selection decisions. *International Conference on Artificial Intelligence in Education*, 226-236.

Mabbott, A., & Bull, S. (2006). Student preferences for editing, persuading, and negotiating the open learner model. *International Conference on Intelligent Tutoring Systems*, 481-490.

Martin, B., & Mitrovic, A. (2002a). Authoring web-based tutoring systems with WETAS. *Computers in Education, 2002. Proceedings. International Conference on*, 183-187.

Martin, B., & Mitrovic, A. (2002b). Wetas: A web-based authoring system for constraint-based its. *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 543-546.

Mathews, M., Mitrovic, A., Lin, B., Holland, J., & Churcher, N. (2012). Do your eyes give it away? Using eye tracking data to understand students' attitudes towards open student model representations. *International Conference on Intelligent Tutoring Systems*, 422-427.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, *63*(2), 81.

Mitrovic, A. (2001). Investigating students' self-assessment skills. *International Conference on User Modeling*, 247-250.

Mitrović, A. (1998). Experiences in implementing constraint-based modeling in SQL-Tutor. *International Conference on Intelligent Tutoring Systems*, 414-423.

Mitrovic, A., Koedinger, K. R., & Martin, B. (2003). A comparative analysis of cognitive tutoring and constraint-based modeling. *International Conference on User Modeling*, 313-322.

Mitrovic, A., & Martin, B. (2002). Evaluating the effects of open student models on learning. *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 296-305.

Mitrovic, A., & Martin, B. (2003). Scaffolding and fading problem selection in SQL-Tutor. *Proceedings of the 11th International Conference on Artificial Intelligence in Education*, 479-481.

Mitrovic, A., & Martin, B. (2007). Evaluating the effect of open student models on self-assessment. *International Journal of Artificial Intelligence in Education*, *17*(2), 121-144.

Mitrovic, A., Martin, B., & Mayo, M. (2002). Using evaluation to shape ITS design: Results and experiences with SQL-Tutor. *User Modeling and User-Adapted Interaction*, *12*(2-3), 243-279.

Mitrovic, A., Mayo, M., Suraweera, P., & Martin, B. (2001). Constraint-based tutors: a success story. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 931-940.

Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language.

Ohlsson, S. (1994). Constraint-based student modeling *Student modelling: the key to individualized knowledge-based instruction* (pp. 167-189): Springer.

Ohlsson, S. (1996). Learning from performance errors. *Psychological review, 103*(2), 241.

Paas, F. G., & Van Merriënboer, J. J. (1993). The efficiency of instructional conditions: An approach to combine mental effort and performance measures. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 35*(4), 737-743.

Paas, F. G., & Van Merriënboer, J. J. (1994). Instructional control of cognitive load in the training of complex cognitive tasks. *Educational psychology review, 6*(4), 351-371.

Payne, S. J., & Squibb, H. R. (1990). Algebra Mal-Rules and Cognitive Accounts of Error. *Cognitive science, 14*(3), 445-481.

Polson, M. C., & Richardson, J. J. (1988). *Foundations of intelligent tutoring systems*: Psychology Press.

Self, J. A. (1990). Bypassing the intractable problem of student modelling. *Intelligent tutoring systems: At the crossroads of artificial intelligence and education, 41*, 1-26.

Self, J. A. (1995). Computational mathetics. *Univ. of Leeds*.

Suraweera, P., & Mitrovic, A. (2002). KERMIT: A constraint-based tutor for database modeling. *International Conference on Intelligent Tutoring Systems*, 377-387.

Suraweera, P., & Mitrovic, A. (2004). An intelligent tutoring system for entity relationship modelling. *International Journal of Artificial Intelligence in Education, 14*(3, 4), 375-417.

Swanson, H. L. (1990). Influence of metacognitive knowledge and aptitude on problem solving. *Journal of educational psychology, 82*(2), 306.

Sweller, J., Van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review, 10*(3), 251-296.

Xie, B., & Salvendy, G. (2000). Prediction of mental workload in single and multiple tasks environments. *International journal of cognitive ergonomics, 4*(3), 213-242.

Young, M. S., & Stanton, N. A. (2002). Malleable attentional resources theory: a new explanation for the effects of mental underload on performance. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 44*(3), 365-375.

Zakharov, K., Mitrovic, A., & Ohlsson, S. (2005). Feedback micro-engineering in EER-Tutor. *Proceedings of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, 718-725.

Zimmerman, B. J. (2008). Investigating self-regulation and motivation: Historical background, methodological developments, and future prospects. *American Educational Research Journal*, *45*(1), 166-183.